LAMP-TR-103
CS-TR-4492
UMIACS-TR-2003-61
CFAR-TR-985

MDA9040-0C-2110
December 2002

An Appearance Based Approach for Human and Object Tracking

Martí Balcells Capellades

Language and Media Processing (LAMP) Laboratory
University of Maryland, College Park, MD 20742-3275
balcells@cfar.umd.edu

Abstract

We have implemented a system for tracking humans and detecting human-object interactions. Persistent tracking of humans and objects in a video sequence is an important task in surveillance applications. Pose and illumination variations, occlusion, appearance and disappearance of humans in the scene, etc... are some of the challenges one has to face. We present an appearance based approach to the problem. A combination of correlogram and histogram information is used to model object and human color distributions. Humans and objects are detected using a background subtraction algorithm. The models are built on the fly and used to track them on a frame by frame basis. The system is able to detect when humans merge into groups and segment them during occlusion. Identities are preserved during all the sequence, even if a person enters and leaves the scene. The system is also able to detect when a person deposits or removes an object from the scene. In the first case the models are used to track the object retroactively in time. In the second case the objects are tracked for the rest of the sequence. The model is able to overcome common deformations as well as many situations involving occlusion. Furthermore, it is easy to update. We assume a static camera and focus on compressed images taken in an indoor environment. The results show that this is a powerful processing technique providing important information to algorithms performing higher level analysis such as activity recognition, where human-object interactions play an important role.

Research partially suported by the Advanced Research and Development Activity (ARDA) under contract number MDA 9040-0C-2110.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
DEC 2002		**00-12-2002 to 00-12-2002**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
An Appearance Based Approach for Human and Object Tracking	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Language and Media Processing Laboratory,Institute for Advanced Computer Studies,University of Maryland,College Park,MD,20742-3275**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**		**78**	

Chapter 1

Introduction

Every day the demand for applications requiring a high level of understanding of video content grows. For instance, in a surveillance application where possible threats or suspicious activities have to be detected, the system is required to be able to automatically answer questions like how many humans are in the scene, the identity of each person, where were these humans before, and with whom or which objects have they interacted and what have they been doing. In a teleconferencing environment similar questions need to be answered if a system is to automatically manage the cameras and decide for example, which scenes should be shown to each person. And there are many other applications (i.e. video retrieval, interface to games, etc...) where the availability of such information is fundamental. The intrinsic complexity of interactions between humans, objects and the world, however, makes the problem very challenging unless some constraints or assumptions are used.

The main objective of this work is to track humans through a video sequence, maintaining continuous labeling even if the person leaves the scene for a long period of time and returns. Simple interactions between humans and objects such as a person removing an object from the scene or depositing a new object in the scene, are detected. All this information can then be provided to systems performing higher level analysis (i.e. activity recognition, object or human recognition, etc...). The system should be able to handle a wide range of video sequences from many different sources, and the algorithms must work with degraded or compressed data. Furthermore, no specific assumptions should be made about the camera angle or position. One main constraint we impose however, is that the video be taken from a static camera. The main reason for assuming this constraint is that the pre-processing stage of the system is a background subtraction algorithm [14] that basically has as output all the regions of the image that are different from a learned background. This module simplifies the complexity of the post-processing stage. Furthermore, before approaching the more difficult moving camera case, the problems that arise in the static camera situation have to be well understood.

The study is focused on indoor sequences, although the system can be adapted for outdoor scenes. It is assumed that the humans are relatively large with respect to the noise introduced by the background subtraction stage. This is a reasonable assumption in indoor sequences, however

it might not be always true for outdoor sequences, where the images often have a wide range of view and the size of the humans or the objects can be very small.

The implemented system has two modules. One is responsible for human tracking. The other is responsible for detecting and tracking objects that have been deposited or removed from the scene. In both scenarios, an appearance based model is used to track humans and objects and segment them under occlusion. This model is based on the so called color correlogram, that basically consists of a co-occurrence matrix that expresses the probability of finding a color c_1 at a distance d of a color c_2 for specified distances.

The human tracking part consist of different stages. In the first stage the background is subtracted and the foreground regions are detected. After cleaning the image, moving blobs larger than a specified size are kept. At this point it is assumed that every blob will correspond to one and only one person. For each foreground blob, the color model is initialized and a distance measure is used to consistently label the blobs for the rest of the sequence. The model is robust in the presence of partial occlusion or pose variations and is updated at every frame. When two blobs merge into one, the color model of each blob is used to classify pixels as belonging to one or the other using a maximum likelihood criterion. A color model is then initialized for the group and used to track it until the components split again.

Detection of human and objects interaction is also based on the background subtracted image. When a blob splits into two blobs and one of them is static, either an object has been deposited or has been removed from the scene. The gradient response is then evaluated around the detected object boundaries and if it is greater in the background frame than in the current frame, it is concluded that the object has been picked up. Otherwise it is assumed to have been deposited. A color model is then initialized for both the object and the person and it is used to segment and track them for the rest of the video sequence, or in the case of a deposited object, to back track and segment the objects in earlier frames.

In order to test the algorithms, several indoor video clips taken from a stationary camera have been recorded. Each sample video has been chosen to show important features of the algorithm. Each sequence is challenging in some aspect. For example there is a sequence with two humans wearing very similar clothes, many with three and five humans forming groups and splitting in several different ways. There is also a sequence with five humans entering and leaving the scene several times to test the verification process. The object tracking module has also been tested under different conditions. There are sequences involving small and big objects, with more or less challenging colors, and there are also examples showing objects being removed and deposited. There will always be situations under which the performance of the algorithm degrades, but the experiments show that the system can handle a wide range of situations and that the correlogram is a powerful tool for appearance-based modeling.

The rest of the thesis is organized as follows. In Chapter 2, a survey of literature related to our work is presented. In Chapter 3, the appearance based model used for tracking is presented. Methods used for updating the model, calculating the distances between two models and segmenting under occluding conditions are also presented. In Chapters 4 and 5, the systems for tracking humans and objects respectively are described and results are discussed. Finally, Chapter 6 presents the summary and conclusions.

Chapter 2

Survey of related work

Over the last several years there has been significant number of efforts [1, 3, 4, 5, 6, 7, 9, 10, 12, 11, 15, 16, 17, 18, 19, 20, 27] reported for detecting and tracking humans. However the problem is still far from being solved. Because of the inherent complexity of the task, each system has made many assumptions and several different approaches have been taken. Therefore each system can be classified according to the assumptions made. There are systems that are designed to work in outdoor environments, others only for indoor scenes. Most of the reported works assume a static camera while some work with both static and moving cameras. The number of cameras (single camera, stereo, or multiple cameras) can also lead to fairly different approaches and the same can be said about the number of humans each system can deal with. While there are systems that can track only one person at a time, others track multiple humans and even handle occlusions and other kind of interactions between humans.

Occlusion is one of the more challenging problems when trying to track humans, mainly in indoor video. In addition to occlusions between humans, there may be many objects occluding humans moving through the scene. Although outdoor video also has the same problems, it is more probable to find an open space in an outdoor scene than in an indoor sequence. One of the options when trying to solve this problem is using multiple cameras. Having multiple views of the same scene, ambiguities caused by occlusions may be resolved. In [1] Mittal and Davis use a multi-perspective video approach to segment, detect and track multiple humans when the scene being viewed is sufficiently crowded. In such scenes it cannot be assumed that any or all of the humans in the scene would be visually isolated from any vantage point. They assume that cameras are frame synchronized and calibrated, and that the humans are moving on a calibrated ground plane. The system is fully automatic and they report fairly good results. In [3] Orwell et al. use several uncalibrated cameras to track multiple objects. The connected blobs belonging to moving objects are obtained using a background subtraction technique. The color distributions from objects are modeled using histograms and the model is used to match and track each object. In [4] Krumm et al. describe a system that is able to track multiple humans in a room using two sets of color stereo cameras mounted on the walls. They use stereo images to locate humans and the color information is used to maintain their identities.

In [5], Intille et al. use a single camera and solve the occlusion problem by locating the camera on the top. The top-down view considerably reduces the ambiguity. In the Pfinder system [6] Wren et al. were mainly interested in recovering a 3D description of a person in a large room and they were able to track head and hands.

Many of the assumptions made by the previous systems [1, 3, 4, 5, 6] would not be acceptable for our application. The objective is to track individuals in video recorded in many different environments, with no control on the position of the camera. As in [7, 9, 10, 11, 12] imagery collected by a single static camera is assumed.

W^4 [7] is a real-time surveillance system for detecting and tracking multiple humans and monitoring their activities in an outdoor environment. It is designed to work with high-quality controlled-gain uncompressed gray scale imagery and employs a combination of shape and motion analysis to locate humans and their body parts (head, hands, feet and torso) and tracks them using appearance based models. W^4 assumes that the whole body is visible and when several humans form a group, the system is able to segment them as long as the head of each person is visible. W^4 is also able to determine whether a person is carrying an object using symmetry constraints and periodicity analysis, and then track the object during exchanges. In [8] Haritaoglu et al. incorporated stereo information into their system.

McKenna et al. [9] also implemented a complete system for tracking multiple humans in a relatively unconstrained environment. They perform tracking at three levels of abstraction, regions (blobs resulting from a background subtraction algorithm), humans and groups. However they are not able to segment individuals when they are in a group. They are also able to detect simple interactions with objects.

In [10] Senior et al. use an appearance based model to track humans through occlusions. The models are used to localize objects during partial occlusions, detect complete occlusions and resolve depth ordering of objects. Fuentes and Velastin [11] present a real-time tracking system that matches the blobs from two consecutive frames using what they call direct and inverse matching matrices and they are able to successfully detect merging and splitting of blobs. In [12] Stauffer and Grimson use a linear predictive multiple hypothese tracking algorithm to match the connected components obtained from a background subtraction algorithm. The system is able to classify objects belonging to two classes, cars or humans, using aspect ratio. However they do not handle occlusion.

Our system, as well as many of the previous systems [7, 9, 10, 11, 12], share the first processing stage, viz., of background subtraction. The main idea behind background subtraction is to model the background so that objects that do not belong to that modeled background can be detected. This model should be adaptive in order to overcome global illumination changes. Other common problems that can arise are local variations on lighting conditions (shadows, highlights, ...) or small movements of background objects (for example the wind moving tree branches). To solve this problem the distribution of each pixel can be modeled. Each of the cited papers uses a different technique to do so. For instance, [4, 5, 6] use a single Gaussian, while [7, 8] use a bimodal distribution and [12] uses a mixture of Gaussians, although this approach considerably increases the computational load. In [9] McKenna et al. use a method for background subtraction that not only is based on color but also on gradient information that helps in removing several types of shadows. Instead of using a background subtraction algorithm, Fuentes and Velastin [11] use a foreground detection algorithm that is based on the luminance contrast together with some

filtering to reduce noise. Senior et al. [10] attempt to make the system robust to illumination changes by modeling the brightness and color distortion in the RGB space, using an approach similar to Horprasert et al. [13]. Horprasert's algorithm is able to cope with local illumination changes due to shadows and highlights, as well as global illumination changes. A problem with all these parametric modeling approaches is that may fail for compressed video, where there is discretized noise and the distribution of the data is not smooth and is non-deterministic. In our system an algorithm developed by Horprasert, Harwood and Kim [14] that employs vector quantization techniques is used. This codebook-based background modeling approach encodes the background scene observed during a very long period of time by compressing the long history of background pixel values in a codebook. In order to eliminate the detection of shadowed or highlighted regions they tolerate some color and brightness distortion at every pixel. The background model is adaptive and can handle the presence of moving foreground objects during the learning process. The method is very robust to the quality of the video and works specially well for MPEG videos. From the users point of view, there are five parameters that have to be set. Two of them are thresholds that control the shadow and highlight detection. Another parameter controls the threshold on the number of times that a pixel has to appear in the learning process in order to be considered part of the background. The remaining two parameters are the more critical ones and are very challenging to set. One controls how different two pixel values have to be in order to be encoded in the same codeword or not. The other one is for the detection process and determines how different a pixel value has to be from the background model in order to be represented by the same codeword. Kim and Harwood have recently developed a method to automatically estimate these parameters although until now it only works for uncompressed video.

The main problem with all the approaches that relay on background subtraction algorithms is that they usually assume a static camera. Even though there have been efforts to extend the same idea to a moving camera by building a panoramic view of the scene and then detecting moving objects [2], for many sequences the results would be still too noisy to be as useful as the results of static camera approaches. A completely different method for the moving camera case can be used instead. In [15] Philomin et al. use a deformable shape model for humans coupled with a variant of the Condensation algorithm [43] that uses an efficient quasi-random sampling strategy for tracking humans from a moving camera with no restrictions on the viewing angle and the dynamics of the camera or the human being tracked.

Once the system has obtained the connected components belonging to each person, a model is needed in order to match the blobs over time or to be able to handle occlusion. There has been a lot of work in sophisticated 3D articulated models for human tracking [16, 17, 18, 19]. The main drawbacks of these approaches are that they are computationally very expensive and usually initialized by hand. For tracking, less specific models are enough and always can be used for initialization and reinitialization of more complex models. Appearance based models have been very popular. Color distributions have been effectively modeled for tracking using both histograms [4, 9, 20] and Gaussian mixture models [9, 21], estimated from color data using an expectation-maximization algorithm. Both models can be updated adaptively.

In the object recognition literature many interesting representation models have been proposed. Swain and Ballard [22] represented objects by their color histogram. Their technique was shown to be remarkably robust to changes in the object's orientation, changes of the scale of the object, partial occlusion or changes in the viewing position. These are all desirable properties for our

purpose. However a main drawback in this technique is that it is very sensitive to lighting conditions. To overcome this problem, Funt and Finlayson [23] used derivatives of the logarithms of the color channels (assuming a locally constant illumination). Finlayson et al. [24] also developed an image normalization method to remove image dependency on lighting geometry and illumination color. In the process however, there is some loss of information that might degrade the recognition performance. Other approaches model objects based on local characteristics. In [25] Schiele and Pentland use a vector of local features measured by local operators such as Gaussian derivatives or Gabor filters. In [26] Schiele and Crowley use multidimensional receptive field histograms, a representation that consists of a probability density function or joint statistics of local appearance as measured by a vector of robust local shape descriptors.

In [27] Elgammal and Davis developed an appearance based model to segment humans under occlusion using a maximum likelihood criterion. The human model was based on segmenting the body into regions (head, torso and legs) and then model color distributions of each region. They used a kernel density estimation method.

Co-occurrence matrices have also been used for many different applications. They were first introduced by Haralick et al. [28] in 1973 to extract textural features for image classification. In [29] they were used for object recognition and matching and in [30, 31] for image retrieval. In fact, color co-occurrence matrices can be seen as a specific kind of geometric histograms, introduced by Rao et al. in [32].

Chapter 3

The appearance based model

3.1 Introduction

In many computer vision problems there is a need to find a representation for objects or entities that are relevant when solving a given problem. For instance, in the face recognition problem, one needs to model the face of each person. Depending on the application the model will have to contain a specific set of characteristics. The model will also have to account for the conditions of the input data. For examlpe, if there are changes in illumination, the data is compressed, the view changes over time, or there are large variations on object sizes, then the model will have to be general enough to handle these variations.

The model for visual appearance proposed here was first developed with the objective of being able to identify humans in an indoor environment. The model should capture relevant information to differentiate between different humans and to recognize humans that have been observed before. Since humans are very complex deformable entities it is difficult to find a single good model that can handle all possible situations. As people can be close to the camera or far away, the model has to be invariant to changes in scale. The model should, to the extent possible, be invariant to rotation and translation. The model should also be robust to partial occlusions, deformations and changes in illumination.

Taking into account all these requirements it seems that the best option is to use a 3D model of a person like the one proposed in [17] by Moon and Chellappa combined with some color cues. A 3D model of a person should be able to handle typical deformations, changes in scale or camera view, occlusions, etc. However the complexity of the model would make the total computational cost too high, not to mention the fact that this kind of 3D model usually needs to be initialized by hand and we want our system to be as automatic as possible.

Therefore we decided to use a 2D model. Since the requirements of our model were very similar to the requirements that arise in the object recognition problem, we searched for possible solutions in the object recognition literature. A possible approach could be to use the Karhunen-Loeve transformation [33] to obtain a compact representation of the objects. Eigen analysis has

been successfully used in applications like face recognition [34], face detection [35] and object recognition [36]. A major drawback of the eigenvector approach is that changes in the individual pixel values, caused, for example by translation, rotation or illumination changes will change the eigenvector representation of the object. Another possible approach is using histograms. In [22] Swain and Ballard identified objects by matching a color histogram from an image region with a color histogram from a sample of the object. The technique was shown to be remarkably robust to changes in object orientation, scale, partial occlusion or changes in viewing positions and even in the shape of the object. These are all desirable properties in our model. Histograms have been also successfully used in people tracking [9] and in image retrieval [22, 37]. A major drawback of this method is its sensitivity to lighting conditions.

There have been many attempts to incorporate spatial information with color. In [38, 39, 40] the image was divided into regions and in [41] Pass used a histogram-refinement approach that appears to perform much better than color histograms. In [31], Huang et al. used color correlograms for image indexing and other applications (image subregion querying, image location, and cut detection). Their experiments showed that this feature was able to outperform both the traditional histogram method and the histogram-refinement method for image indexing/retrieval. Therefore we concluded that color correlograms could be a good model for our purposes too. Since models will be updated at every frame, illumination conditions would not significantly affect the performance of our algorithm. In addition we could also use the image normalization technique introduced in [24].

3.2 The model: the color correlogram

A color correlogram [31] is an image feature that expresses the probability that, given a pixel of a color c_i, we find a pixel of color c_j at a distance d. It expresses how the spatial correlation of pairs of colors change with distance. A color histogram captures only the color distribution in an image and does not include any spatial correlation information.

Notation. Let $\mathcal{I}$ be an $n_1 \times n_2$ image, quantized into m colors $c_1, ..., c_m$. For a pixel $p = (x, y) \in \mathcal{I}$, let $\mathcal{I}(p)$ denote its color. Let $\mathcal{I}_c \equiv \{p | \mathcal{I}(p) = c\}$. Thus, the notation $p \in \mathcal{I}_c$ is synonymous with $p \in \mathcal{I}$, $\mathcal{I}(p) = c$. For convenience we use the L_∞-norm to measure the distance between pixels, i.e., for pixels $p_1 = (x_1, y_1)$, $p_2 = (x_2, y_2)$, we define $|p_1 - p_2| \equiv max\{|x_1 - x_2|, |y_1 - y_2|\}$. We denote the set $1, 2, ..., n$ by $[n]$.

Histogram. The color histogram h of $\mathcal{I}$ is defined for $i \in [m]$ such that $h_\mathcal{I}(c_i)$ gives for any pixel in $\mathcal{I}$, the probability that the color of the pixel is c_i. Given the count

$$H_{c_i}(\mathcal{I}) \equiv |\{p \in \mathcal{I}_{c_i}\}| \tag{3.1}$$

it follows that

$$h_\mathcal{I}(c_i) = \frac{H_\mathcal{I}(c_i)}{|\mathcal{I}|} \tag{3.2}$$

Correlogram. Let a distance set $D \subseteq [min\{n_1, n_2\}]$ be fixed a priori. Let $d = |D|$. Then the correlogram of $\mathcal{I}$ is defined for $i, j \in [m], k \in D$ such that $\gamma_\mathcal{I}(c_i, c_j, k)$ gives the probability that a pixel at a distance k from a given pixel of color c_i is of color c_j. The size of the correlogram is

m^2d. If we define the following count

$$\Gamma_{\mathcal{I}}(c_i, c_j, k) \equiv |\{p_1 \in \mathcal{I}_{c_i}, p_2 \in \mathcal{I}_{c_j} | |p_1 - p_2| = k\}| \tag{3.3}$$

then,

$$\gamma_{\mathcal{I}}(c_i, c_j, k) = \frac{\Gamma_{\mathcal{I}}(c_i, c_j, k)}{8kH_{\mathcal{I}}(c_i)} \tag{3.4}$$

The *autocorrelogram* of $\mathcal{I}$ captures the spatial correlation between identical colors only and is defined by

$$\alpha_{\mathcal{I}}(c_i, k) \equiv \gamma_{\mathcal{I}}(c_i, c_i, k) \tag{3.5}$$

This requires only md space. While choosing d to define the correlogram, we need to address the following issue. A large d would require expensive computation and large storage requirements. A small d might compromise the quality of the feature. In general, since local correlations between colors in an image are more significant than global correlations, a relatively small value of d is sufficient to capture the spatial correlation.

3.3 Comparing feature vectors: distance measures

Once we have a representation for objects, there is the need to find a good similarity measure between feature vectors that allows us to classify an observation given a set of models. This measure is commonly determined by some kind of distance between the corresponding descriptors in feature space. There are many different distance measures that one can use. L_1 and L_2 norms have been commonly used, and depending on the application one or the other performs better. The L_1 norm is definded as $L_1(x, y) = \sum_{i=1}^{n} |x_i - y_i|$ where x and y are n dimensional vectors. The L_2 norm is defined as $L_2(x, y) = \sqrt{\sum_{i=1}^{n} |x_i - y_i|^2}$. There are other distances one can use. Hafner et al. [42] suggested using a more sophisticated form of quadratic distance measure that tries to capture perceptual similarity between any two colors. They also proposed a method to avoid the computational expense of a quadratic distance measure by prefiltering with a simpler distance, and then using the quadratic L_2 distance for a smaller set of feature vectors. In [31], Huang et al. used a normalized L_1 distance measure for comparing histograms and correlograms because it is simple and statistically more robust to outliers than the L_2 measure.

We use a slightly different version of the normalized L_1 distance. Given two images $\mathcal{I}$ and $\mathcal{I}'$, their dissimilarity can be computed using the following formulas:

$$D_h(\mathcal{I}, \mathcal{I}') \equiv |\mathcal{I} - \mathcal{I}'|_h \equiv \frac{\sum_{\forall i \in [m]} |h_{\mathcal{I}}(c_i) - h_{\mathcal{I}'}(c_i)|}{\sum_{\forall j \in [m]} h_{\mathcal{I}}(c_j) + h_{\mathcal{I}'}(c_j)} \tag{3.6}$$

$$D_\gamma(\mathcal{I}, \mathcal{I}') \equiv |\mathcal{I} - \mathcal{I}'|_\gamma \equiv \frac{\sum_{\forall i,j \in [m], k \in [d]} |\gamma_{\mathcal{I}}(c_i, c_j, k) - \gamma_{\mathcal{I}'}(c_i, c_j, k)|}{\sum_{\forall i,j \in [m], k \in [d]} \gamma_{\mathcal{I}}(c_i, c_j, k) + \sum_{\forall i,j \in [m], k \in [d]} \gamma_{\mathcal{I}'}(c_i, c_j, k)} \tag{3.7}$$

The measure itself is symmetric ($|\mathcal{I} - \mathcal{I}'|_h = |\mathcal{I}' - \mathcal{I}|_h$) and lies within $[0, 1]$. The subscripts h and γ mean histogram and correlogram respectively. The corresponding similarity measure can be simply derived from $D(\mathcal{I}, \mathcal{I}')$ as

$$S_h(\mathcal{I}, \mathcal{I}') \equiv 1 - D_h(\mathcal{I}, \mathcal{I}') \tag{3.8}$$

and

$$S_\gamma(\mathcal{I}, \mathcal{I}') \equiv 1 - D_\gamma(\mathcal{I}, \mathcal{I}') \tag{3.9}$$

3.4 Updating the model

Once the model has been initialized it can be used for tracking and recognition. These applications imply comparing observations with the models stored in a database. If there are significant differences between the objects in the database and the observations, matching might not be possible. These differences might occur due to changes in illumination, pose, or occlusion, between the moment when the model was built and the moment when the observation is made. Therefore a desirable property for a model is that it should be easy to update.

Model updating can be done in two ways. If a stationary distribution is assumed, the model should be updated cumulatively. Otherwise, the model should be updated adaptively. In our case it is more appropriate to update the model adaptively, since the model should adapt to changes that may occur. The updating is done in the following way:

$$h_\mathcal{I}(c_i, t) = \alpha h_\mathcal{I}(c_i, t-1) + (1-\alpha) h_\mathcal{I}^{new}(c_i, t) \tag{3.10}$$

and

$$\gamma_\mathcal{I}(c_i, c_j, k, t) = \alpha \gamma_\mathcal{I}(c_i, c_j, k, t-1) + (1-\alpha) \gamma_\mathcal{I}^{new}(c_i, c_j, k, t) \tag{3.11}$$

where $h_\mathcal{I}^{new}(c_i, t)$ and $\gamma_\mathcal{I}^{new}(c_i, c_j, k, t)$ are the histogram and correlogram respectively computed using only the new image obtained at time t, $h_\mathcal{I}(c_i, t-1)$ and $\gamma_\mathcal{I}(c_i, c_j, k, t-1)$ are the stored models at time $t-1$ and $h_\mathcal{I}(c_i, t)$ and $\gamma_\mathcal{I}(c_i, c_j, k, t)$ are the updated models at time t. α is a constant ($0 \leq \alpha \leq 1$) that determines the velocity of the updating process. If α takes values close to 1, the new information will be slowly incorporated in the model. If α takes values close to 0, the old information will be forgotten rapidly. In general $\alpha = 0.9$ works fairly well.

3.5 Using the model for segmentation

The segmentation problem is described as follows. Given an image and a set of models, for each pixel decide the most probable model it can belong to. This will allow us to estimate people's position when they merge into a group or to segment an object that a person has picked up. There are different ways in which this problem can be addressed. In [27], Elgammal and Davis used a kernel density estimation approach to model the appearance of people and its spatial distribution, and then used a likelihood maximization approach for segmentation.

In the system explained here, two different methods for segmentation have been implemented and tested. One is based on Swain's work [22] with the improvements introduced in [31]. It is explained in Section 3.5.1 and the final criterion for segmentation is expressed in (3.17). The other method is explained in Section 3.5.2 and uses the correlogram probability estimates to do the segmentation based on a Maximum A Posteriori (MAP) criterion. Equation (3.25) shows the final expression used for segmentation.

3.5.1 Histogram backprojection method with correlogram correction for segmentation

In [22] Swain and Ballard proposed a technique for image localization (can also be used for segmentation) that they called the *histogram backprojection* method. The problem is stated as: given a query image $\mathcal{Q}$ and an image $\mathcal{I}$ where $\mathcal{Q} \subseteq \mathcal{I}$, the location in $\mathcal{I}$ where $\mathcal{Q}$ is more likely to be must be found. They use histograms as image representations and they define the following likelihood measure that they call *ratio histogram*:

$$\pi_{c_i,h}(\mathcal{I}|\mathcal{Q}) \equiv min\left\{\frac{H_{\mathcal{Q}}(c_i)}{H_{\mathcal{I}}(c_i)}, 1\right\} \tag{3.12}$$

where $\pi_{c_i,h}(\mathcal{I}|\mathcal{Q})$ is the likelihood that a pixel of color c_i of image $\mathcal{I}$ belongs to the query image $\mathcal{Q}$. $H_{\mathcal{I}}(c_i)$ is the count defined in (3.1). To find the most likely location of the query image the pixel p that maximizes the sum of contributions from all pixels belonging to the subimage of $\mathcal{I}$ centered at pixel p, $\mathcal{I}|p$, has to be found. That is, the location s is given by:

$$s = arg\ max_{p\in\mathcal{I}}\Pi_p(\mathcal{I}|\mathcal{Q}) \tag{3.13}$$

where

$$\Pi_p(\mathcal{I}|\mathcal{Q}) \equiv \sum_{q\in\mathcal{I}|p} \pi_{\mathcal{I}(q),h}(\mathcal{I}|\mathcal{Q}) \tag{3.14}$$

Note that this method gives the same likelihood to all pixels of the same color. It also gives more weight to colors that are frequent in the query but not in the image, thus leading to errors in some cases. To avoid this kind of error, Huang et al. [31] added to (3.14) a factor that they call *correlogram correction*. Since histograms do not have any spatial information, by incorporating a measure coming from the correlogram some local spatial correlation is incorporated.

For each pixel $p \in \mathcal{I}$ the local correlogram is computed at each distance $k \in D$ considering only p and its neighbors (the distance set D should contain relatively small value so that it captures local information for each pixel). Then the correlogram contribution of p is the similarity measure in (3.9) between the local correlogram at pixel $p \in \mathcal{I}$ and a part of correlogram for $\mathcal{Q}$, the query image, that corresponds to color $\mathcal{I}(p)$:

$$\pi_{p,\gamma}(\mathcal{I}|\mathcal{Q}) \equiv S_\gamma(\mathcal{Q}_{\mathcal{I}(p)}, \{p\}) \tag{3.15}$$

where $\pi_{p,\gamma}(\mathcal{I}|\mathcal{Q})$ is the correlogram contribution and $\{p\}$ represents a pixel p along with its neighbors (the neighborhood of p depends on the set of distances where the correlogram has been

computed). The final expression for the likelihood measure from (3.14) is:

$$\Pi_p(\mathcal{I}|\mathcal{Q}) \equiv \sum_{q \in \mathcal{I}|p} \Big(\beta \pi_{\mathcal{I}(q),h}(\mathcal{I}|\mathcal{Q}) + (1-\beta)\pi_{q,\gamma}(\mathcal{I}|\mathcal{Q})\Big) \tag{3.16}$$

where $0 \leq \beta \leq 1$. Usually a value of $\beta = 0.5$ works well.

The likelihood measure in (3.16) can be used to segment humans under occluding conditions. Pixel p in blob $\mathcal{G}$ will be labeled as belonging to model $\mathcal{M}_m$ if and only if:

$$m = arg\ max_i \Pi_p(\mathcal{G}|\mathcal{M}_i) \tag{3.17}$$

3.5.2 A Maximum A Posteriori criterion for segmentation

McKenna et al. [9] addressed the segmentation problem in the context of human tracking. They generalized the ratio histogram idea presented by Swain et al. in [22] to multiple models. They modeled each person using a color histogram and when humans merged, they used a MAP criterion to do the segmentation using the probabilities estimated by the histogram. This approach has the same problem as that of the histogram backprojection method. This probabilities of a pixel belonging to one model only depends on the color of the pixel and not on its position. To incorporate some kind of spatial information to it an equivalent approach to the one proposed by McKenna [9] can be followed, but instead of using the histogram probability estimates, correlogram probability estimates can be used. Using the MAP criterion, a pixel p is classified as belonging to model $\mathcal{M}_m$ if

$$m = arg\ max_i P(\mathcal{M}_i|\{p\}) \tag{3.18}$$

Thus, if given the observation $\{p\}$ (the pixel along with its neighbors), it maximizes the probability of belonging to $\mathcal{M}_m$. Using the Bayes theorem,

$$P(\mathcal{M}_i|\{p\}) = \frac{P(\mathcal{M}_i \cap \{p\})}{P(\{p\})} = \frac{P(\{p\}|\mathcal{M}_i)P(\mathcal{M}_i)}{\sum_{\forall j \in \mathcal{G}} P(\{p\}|\mathcal{M}_j)P(\mathcal{M}_j)} \tag{3.19}$$

where $P(\{p\}|\mathcal{M}_i)$ is the probability of having the neighborhood $\{p\}$ given that belongs to model $\mathcal{M}_i$. To estimate the priors $P(\mathcal{M}_i)$, the area of each person before joining the group $\mathcal{G}$ can be used:

$$P(\mathcal{M}_i) = \frac{|\mathcal{M}_i|}{\sum_{\forall j \in \mathcal{G}} |\mathcal{M}_j|} \tag{3.20}$$

Equation (3.19) can be further expanded using:

$$P(\{p\}|\mathcal{M}_i) = \sum_{\forall c_y \in \{p\}} \sum_{\forall d \in D} P(c_x, c_y, d|\mathcal{M}_i) \tag{3.21}$$

where $P(c_x, c_y, d|\mathcal{M}_i)$ expresses the probability of finding a color c_x at a distance d of a color c_y, given that this pixel belongs to model $\mathcal{M}_i$. This information can be easily found in the correlogram. From the definition,

$$\gamma_{\mathcal{M}_i}(c_x, c_y, d) = P(c_y|c_x, d, \mathcal{M}_i) \tag{3.22}$$

Then,

$$P(c_x, c_y, d|\mathcal{M}_i) = P(c_y|c_x, d, \mathcal{M}_i)P(d|c_x, \mathcal{M}_i)P(c_x|\mathcal{M}_i) \tag{3.23}$$

where $P(c_x|\mathcal{M}_i) = h_{\mathcal{M}_i}(c_x)$ and $P(d|c_x, \mathcal{M}_i) = P(d)$. The priors $P(d)$ are difficult to estimate and so are assumed uniform. Finally, substituting all the expressions into (3.18) and simplifying, it is found that a pixel p will be classified as belonging to model $\mathcal{M}_m$ if and only if:

$$m = arg\ max_i \frac{\sum_{\forall c_y \in \{p\}} \sum_{\forall d \in D} \gamma_{\mathcal{M}_i}(c_x, c_y, d) h_{\mathcal{M}_i}(c_x) P(\mathcal{M}_i)}{\sum_{\forall n} \sum_{\forall c_y \in \{p\}} \sum_{\forall d \in D} \gamma_{\mathcal{M}_n}(c_x, c_y, d) h_{\mathcal{M}_n}(c_x) P(\mathcal{M}_n)} \tag{3.24}$$

To have a smooth segmentation, the posterior probabilities of the neighboring pixels in a specific window W can be added at each pixel location. Then,

$$m = arg\ max_i \sum_{\forall p \in W} \frac{\sum_{\forall c_y \in \{p\}} \sum_{\forall d \in D} \gamma_{\mathcal{M}_i}(c_x, c_y, d) h_{\mathcal{M}_i}(c_x) P(\mathcal{M}_i)}{\sum_{\forall n} \sum_{\forall c_y \in \{p\}} \sum_{\forall d \in D} \gamma_{\mathcal{M}_n}(c_x, c_y, d) h_{\mathcal{M}_n}(c_x) P(\mathcal{M}_n)} \tag{3.25}$$

Chapter 4

Human tracking

4.1 Introduction

Tracking has been widely addressed in the computer vision community since it can provide valuable information. The problem, however, is inherently complex and one needs to put some constraints in order to realize practical solutions.

In our application, input data will be mainly indoor sequences taken from a static camera. We assume that the camera can be in any position or angle but it is assumed that the images will be close enough to the scene so that there will be sufficient pixels on the people. The system should be able to deal with occlusions (including self occlusions), whether they are due to background objects or other people in the scene. People can have any pose and can be entering or leaving the scene. The system has to work for both uncompressed and compressed data. In Figure 4.1 sample frames showing some of the challenges faced by the system are shown. In Figure 4.2 a diagram of the algorithm is shown.

4.2 The algorithm

The very first problem to be solved when tracking people is detecting them. For the static camera case, detection is accomplished by using some kind of *background subtraction* method. Background subtraction consists of modeling the background scene when humans are not present and then comparing every frame with the modeled background. If a person comes in, some background pixels will be occluded and this will be detected by the background subtraction algorithm. Although the idea is simple, the problem is very challenging. Changes in illumination conditions, light movements of objects (for instance bushes moved by the wind), camera noise, shadows or reflections can all effect the detection performance.

We use the background subtraction algorithm designed by Horprasert, Kim, and Harwood [14]. After the background is subtracted, several post-processing to clean up the image. First, small

(a) (b)

(c) (d)

Figure 4.1: Sample frames showing some of the challenges in people tracking: different camera angles, irregular poses and occlusions are illustrated

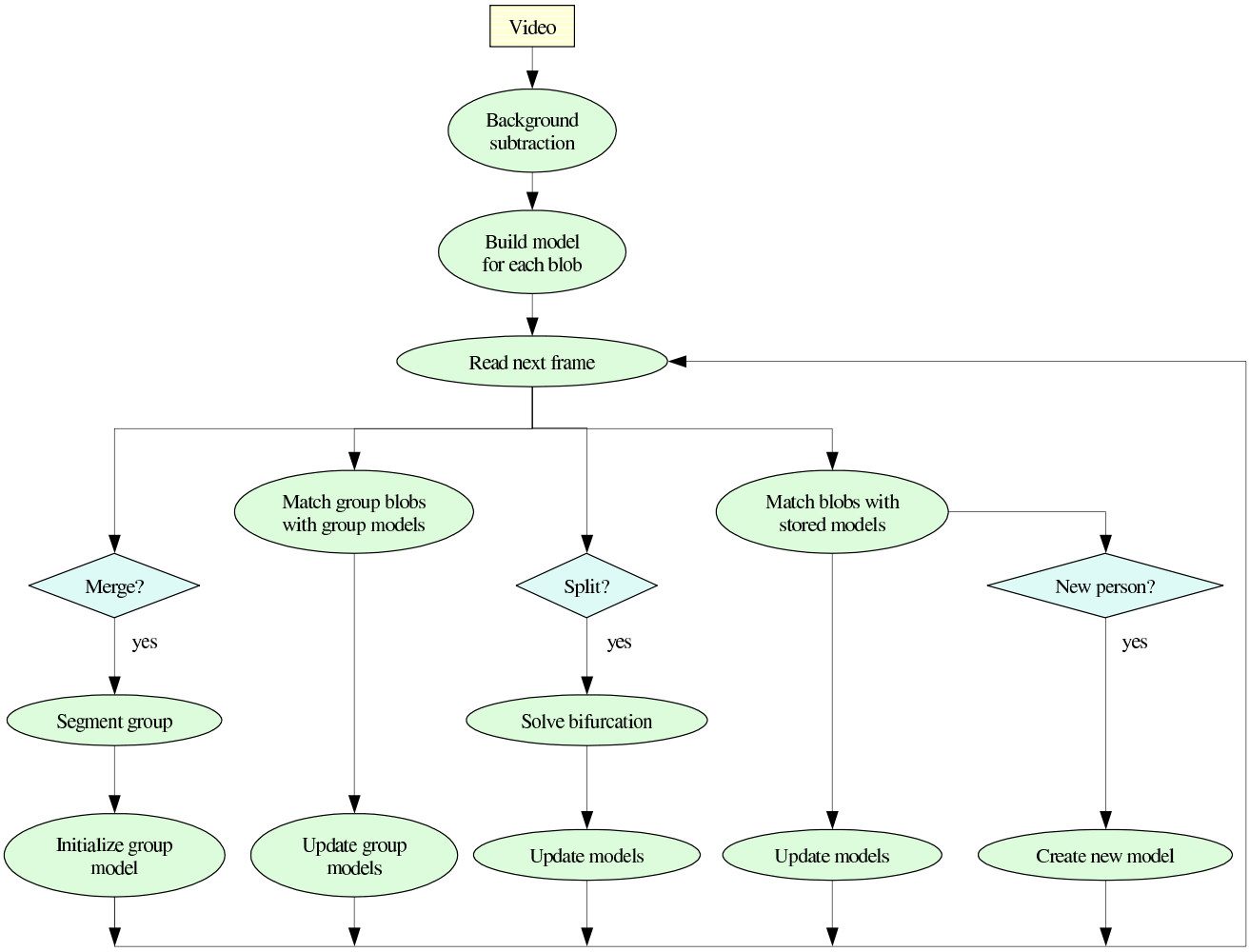


Figure 4.2: The flowchart of human tracking algorithm.

spots (pixels with less than 3 neighbors) are removed. Then a color-based shrink/expand method is applied to foreground boundary pixels. There are three steps: shrink by one pixel, expand by two, and then shrink by one again. The objective of this stage is to fill small gaps and holes in the foreground blobs and to connect nearby blobs. After this stage small blobs are filtered. Subsequently color-based hole filling procedure is applied. A pixel in a hole is filled if it is connected by a path with continuous color similarity. Finally to get smooth silhouettes, boundaries are filtered by opening (erosion followed by dilation) using a rolling ball algorithm [44].

After this processing stage it is assumed that each person will be segmented into one and only one blob. Since in most of the input sequences people are relatively big with respect to the image size, this will usually be the case. It can happen, however, that an object is occluding a person in such a way that the body gets cut into two blobs. In this case the system may conclude that each blob corresponds to a different person. If there is a small part of the body that has been segmented into a different blob after the background subtraction stage, it is assumed that it would be removed during the thresholding process. This will not affect the performance of the tracker since the appearance based model is robust enough to handle these situations. It is also assumed that when a person enters the scene he/she will be isolated. If two people enter together and are segmented into the same blob by the background subtraction algorithm, the system will treat them as though they were a single person. In practice we can adapt the system to model various combinations of people and detect when they separate, generating the model retroactively.

The first time people enter the scene, a model for each of them is stored. As explained in Chapter 3, the appearance based model consists of histogram and correlogram information. In the next frame, another model is built for each of the foreground blobs. Then the similarity (see (3.8) and (3.9)) between all the stored models and all the models of blobs in the current frame is calculated. The most similar blobs are matched as long as the similarity is above certain threshold. However, if a blob in the current frame has a very large distance from all the stored models, a new model is initialized. Empirically the threshold (that can take values between 0 and 1) was set to 0.55. The threshold can also be learned from the sequence itself and set automatically by analyzing the distances between the different realizations of the same model and the distances between other models. The method to automatically set the threshold, however, was not studied in depth and in general, better results are obtained when the threshold is set by hand. Once the matching is done, all the models are updated using (3.10) and (3.11) and the next frame is processed.

Of course, people can form groups that the background subtraction algorithm will segment into a single blob. It is desirable to be able to track these people even when they are in the group or are partially occluded by other people or objects. The system detects that two or more people have merged into a group when the total number of blobs in the frame has decreased and two or more blobs in the previous frame overlap with a single blob in the current frame (see Figures 4.3) and 4.4. When a merge is detected a model for the group is initialized. This model consists of histogram and correlogram of the group, but it also contains the models of each of the people that form the group. The correlogram of the group is used to track the group for the rest of the sequence in the same way that the other blobs are tracked. The models of the components of the group are used to segment the group into each of the people that form it, classifying each pixel as belonging to one of the models using the method explained in Section 3.5. If a person joins an existing group, a new component is added to the group.

Figure 4.3: Sample frames showing a merge between two people. The left image shows the frame before the merge and the right image shows the scene after the merge.

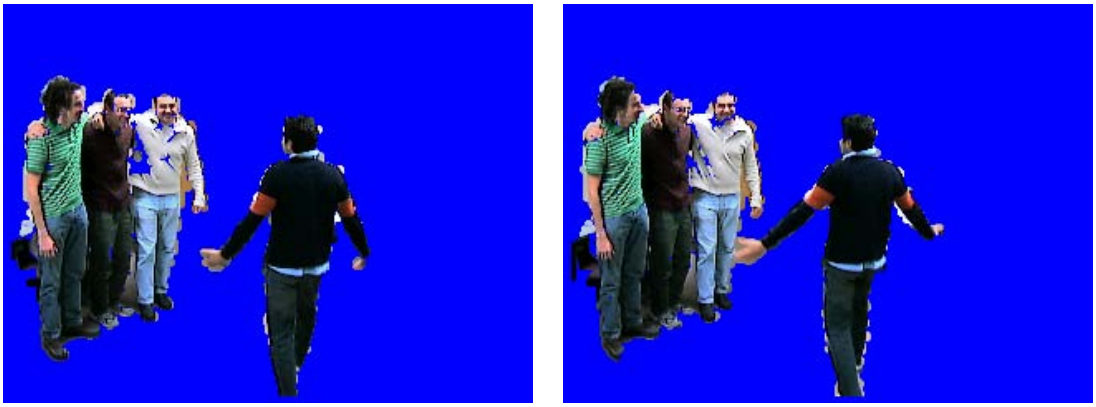


Figure 4.4: Sample frames showing a person merging a group of people. The left frame shows the scene before the merge and the right frame shows the scene after the merge.

Just as several people can merge into a group, the group can also be split. This event is detected when the total number of blobs in the frame has increased and several blobs in the current frame overlap with a group blob in the previous frame (see Figures 4.5 and 4.6). Of course a group can split in many different ways. For instance, a group of four people can split into a group of three persons and a single person or into two groups of two persons. Let $b_1, b_2, ..., b_n$ be the blobs in which the group has split. In order to determine in which blob b_i each person is present, the segmentation is done for all blobs $b_1, b_2, ..., b_n$ using the models of all persons belonging to the group. Let $N(M_i|b_j)$ be the count on the number of pixels in blob b_i that have been classified as model M_i. Then it is concluded that person M_i is present in blob b_j if and only if:

$$N(M_i|b_j) > N(M_i|b_l) \qquad \forall l \neq i \tag{4.1}$$

Figure 4.5: Sample frames showing a split of a two people group. The left column shows the frame before the group splits and the right column shows the frames after the split.

Since segmentation is not perfect, during occlusion the individual models of each of the components of the group are not updated. If they stay together for a long period of time, the illumination conditions may change and the stored models might become inaccurate. In order to minimize these effects, the images are normalized. The comprehensive normalization method proposed in [24] was tested but the results turned out to be worse than when normalization was not used. The loss of information during the normalization method is likely to be high. Therefore a more standard normalization method was used. Instead of using the $< R, G, B >$ pixels themselves, the transformed $< r, g, b >$ were used, where:

$$r = \frac{R \times 255}{R + G + B} \tag{4.2}$$

$$g = \frac{G \times 255}{R + G + B} \tag{4.3}$$

$$b = \frac{B \times 255}{R + G + B} \tag{4.4}$$

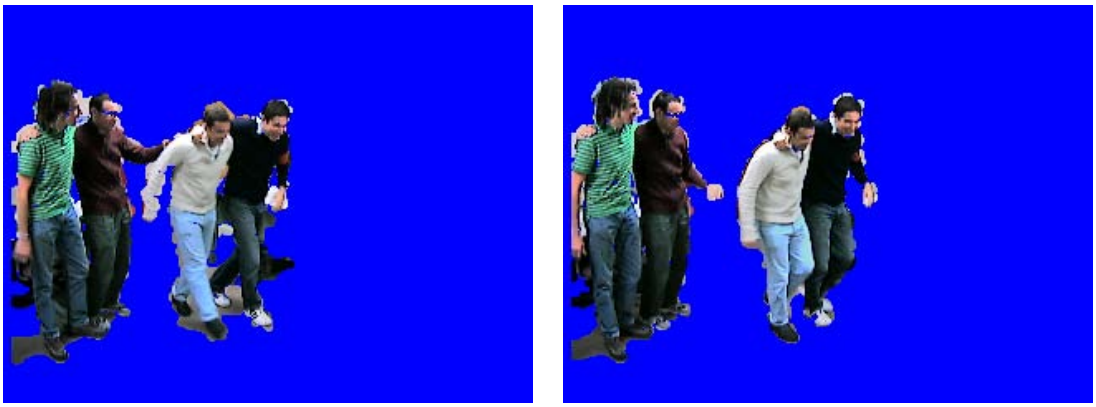


Figure 4.6: Sample frames showing a split of a group of four people into two groups of two people each. The left frame shows the scene before the group splits and the right frame shows the scene after the split.

It can be observed, however, that $b = 255 - r - g$. Since b does not add any new information, people usually discard it. However then there is again a loss in information that could compromise the system, since the original $< R, G, B >$ values of the pixels cannot be recovered. Therefore b was set to

$$b = \frac{R + G + B}{3} \tag{4.5}$$

In general this normalization method helps to improve the results. In some experiments, however, the normalization decreased the performance of the system.

4.3 Results

In order to test the algorithm for tracking humans, several indoor video sequences were recorded. The algorithm has been tested on compressed and uncompressed clips containing different numbers of persons and having different camera positions. In the following sections results are reported for different sequences. In each experiment there is a different challenging aspect that is tested. In all experiments, the images were quantized to 512 colors and the correlogram was calculated at distances $\{1, 3, 5, 7, 10, 15, 23, 35\}$, unless stated otherwise. For all the figures shown in the results, the first column shows the input sequence. The second column shows the frame after background subtraction. The third column shows the final results.

4.3.1 Testing the algorithm and its dependency on several parameters

In this experiment the intention is to test the performance of the algorithm in a general situation. The sequence consists of three persons moving in a room. Individuals form groups and split several times and there are some occluding situations as well. First the algorithm is tested using the histogram backprojection method with correlogram correction described in Section 3.5.1 for segmentation. The results can be seen in Figures 4.7, 4.8 and 4.9. The sequence has a total of 774 frames collected at a rate of approximately 10-15 frames per second. Figures 4.10, 4.11 and 4.12 show the results when the segmentation is done using the MAP criterion described in Section 3.5.2. Although the MAP criterion is supposed to be optimum, the results are a little bit better for the histogram backprojection method with correlogram correction included. This is probably due to the fact that the densities are not estimated properly, or that the assumptions about the priors did not hold. In general, however, the results are good. The identities are preserved during the whole sequence, and segmentation results during occlusion are fairly good. One can see that the pants of two of the individuals have very similar colors and that they are segmented properly during occlusion. It can also be observed that sometimes the algorithm misclassifies the head region. We believe that this is due to the brightness of the skin and the large changes in illumination that occurs.

In some frames parts of the background appear attached to the people, due to shadows that the background subtraction algorithm could not eliminate. When this happens (see for instance Figure 4.8 or Figure 4.9) these regions are segmented as any other region of the group, although they really do not belong to any of the models.

The dependency of the algorithm with respect to different parameter has also been tested. In Figures 4.13, 4.14 and 4.15, the results when the input images are quantized to 1000 colors are shown. In Figures 4.16, 4.17 and 4.18 the results are for images quantized to 64 colors. As it can be observed, the number of colors does not seem to affect the performance of the algorithm and the quality of the segmentation significantly. However, the algorithm is sensitive to the clothing of each individual. Similar observations can be made about the distance set where the correlogram is calculated. In Figures 4.19, 4.20 and 4.21 the results are shown when the correlogram is calculated at distances $\{1, 3, 5, 7\}$. In all cases, the identities are preserved.

4.3.2 Testing split and merge behavior with five humans

In this clip the performance of the algorithm handling splitting and merging between people is tested in a relatively long sequence (1405 frames, approximately 1 minute). The sequence consists of five humans moving in a room, forming groups of up to four people, that split in different ways. For example in Figure 4.22 a group of four persons splits into a group of three persons and a single person, whereas in Figure 4.24 a group of four individuals splits into two groups of two people each. There is also a person leaving the scene for a while and then entering again. All identities are preserved. The images were quantized to 512 colors, and the correlogram was calculated at distances $\{1, 3, 5, 7, 10, 15, 23, 35\}$. For segmentation the histogram backprojection method with correlogram correction was used (Section 3.5.1). The results are shown in Figures 4.22, 4.23 and 4.24. The more people in a group the more difficult the segmentation is. Pants of the people, for example, have similar colors that are challenging to segment. Furthermore, when several

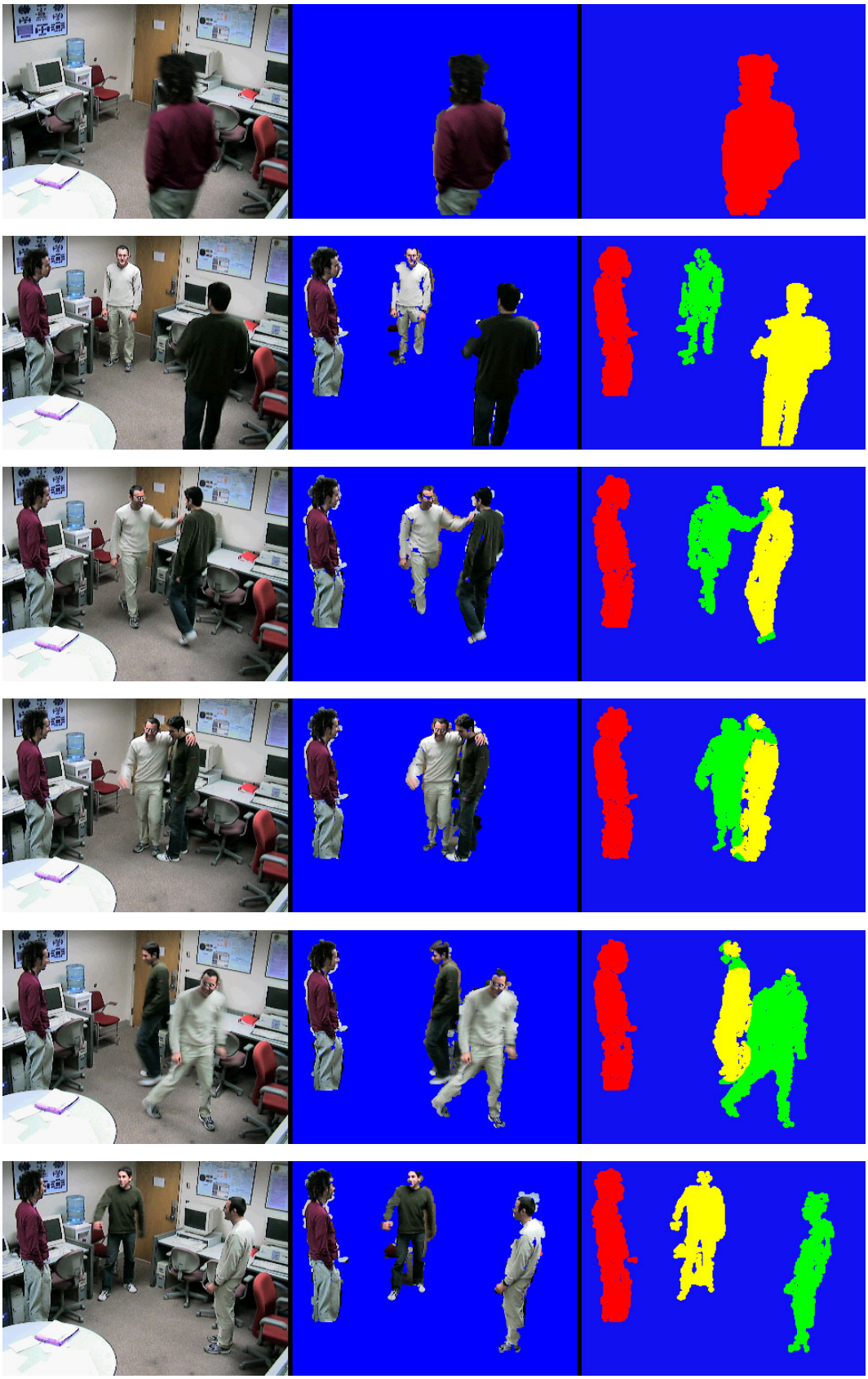


Figure 4.7: A sequence with three persons. Sample frames: 112, 293, 322, 341, 374 and 408. Histogram backprojection method with correlogram correction was used.

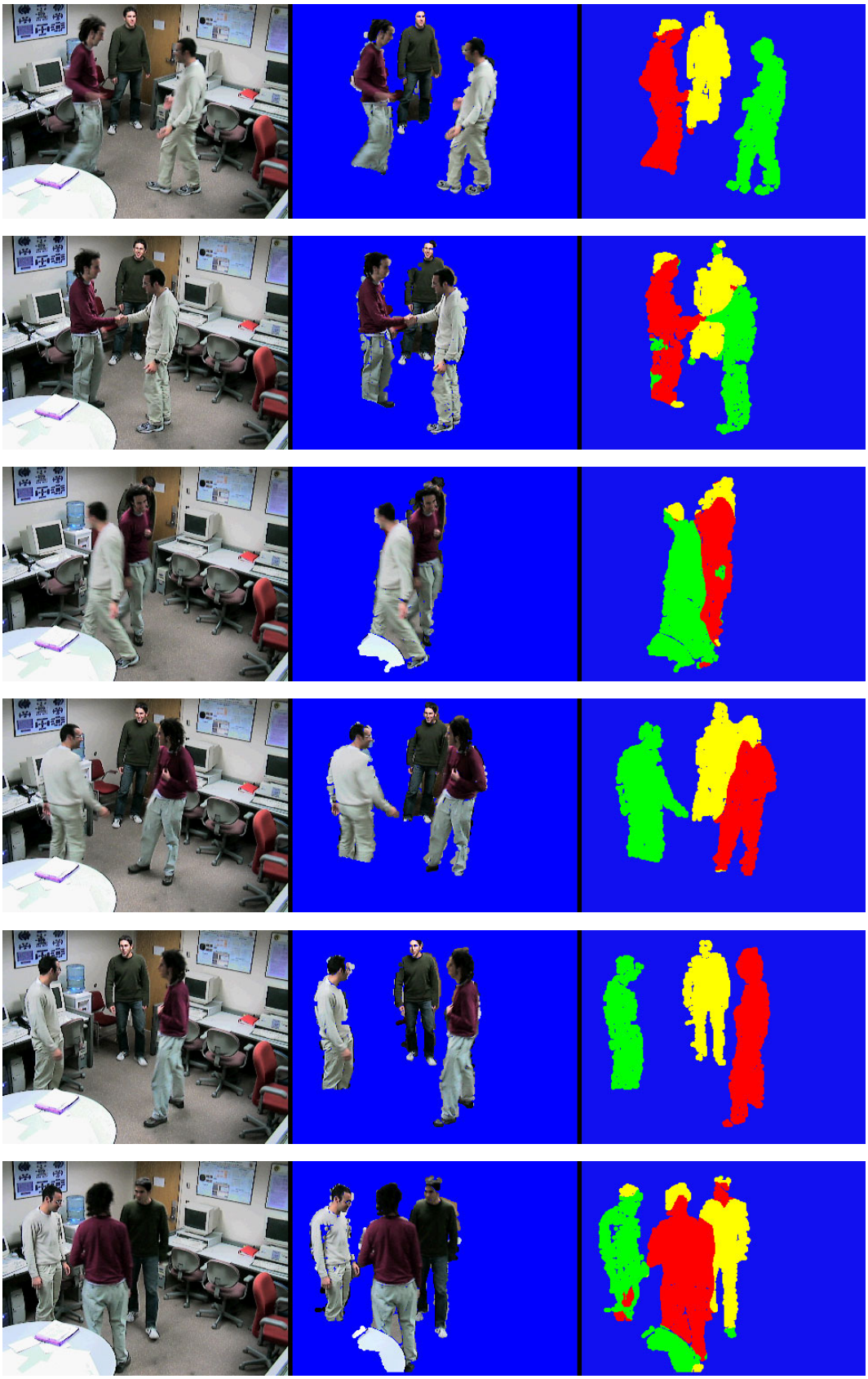


Figure 4.8: A sequence with three persons. Sample frames: 443, 466, 488, 502, 524 and 556. Histogram backprojection method with correlogram correction was used.

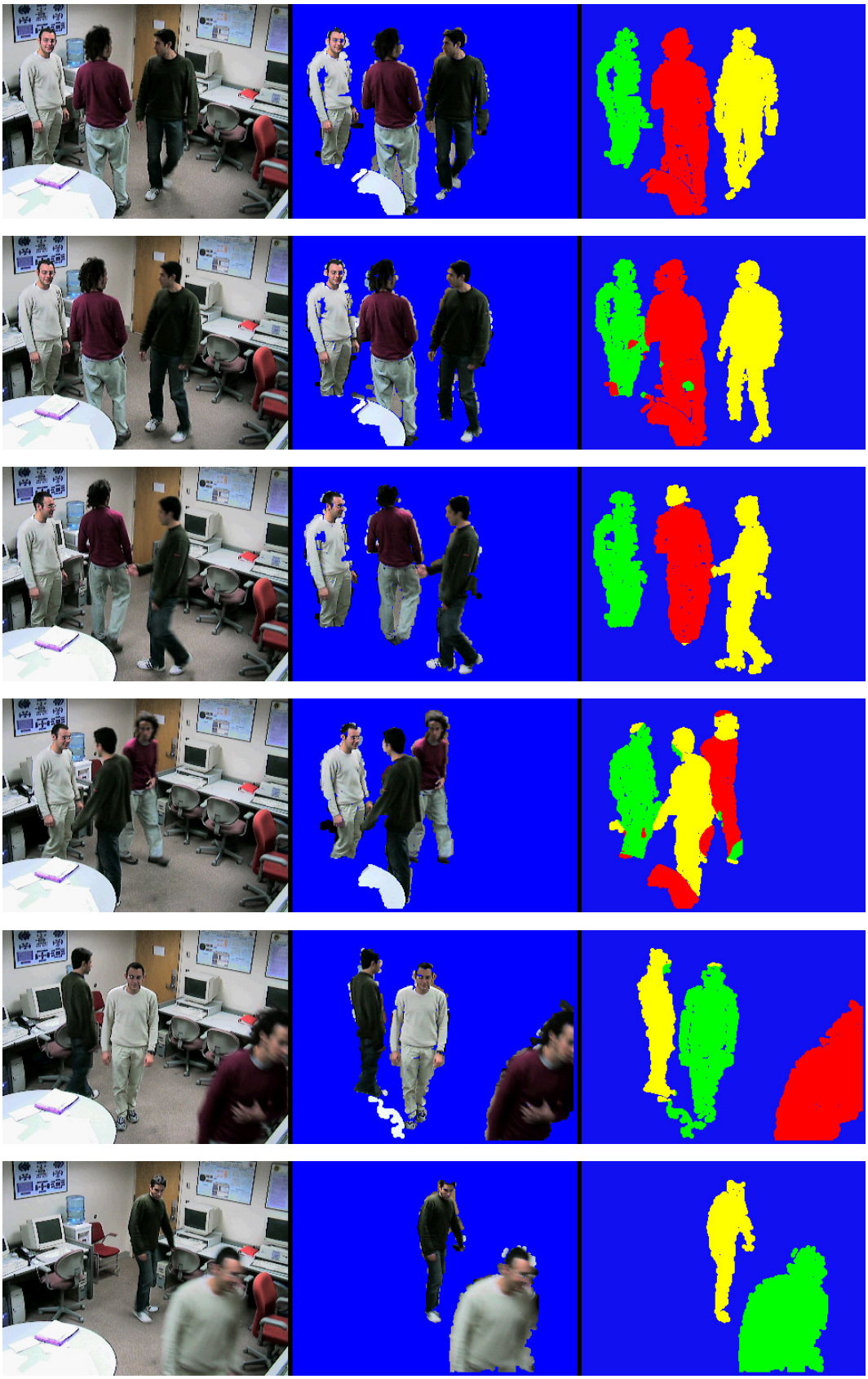


Figure 4.9: A sequence with three persons. Sample frames: 578, 582, 606, 626, 684 and 720. Histogram backprojection method with correlogram correction was used.

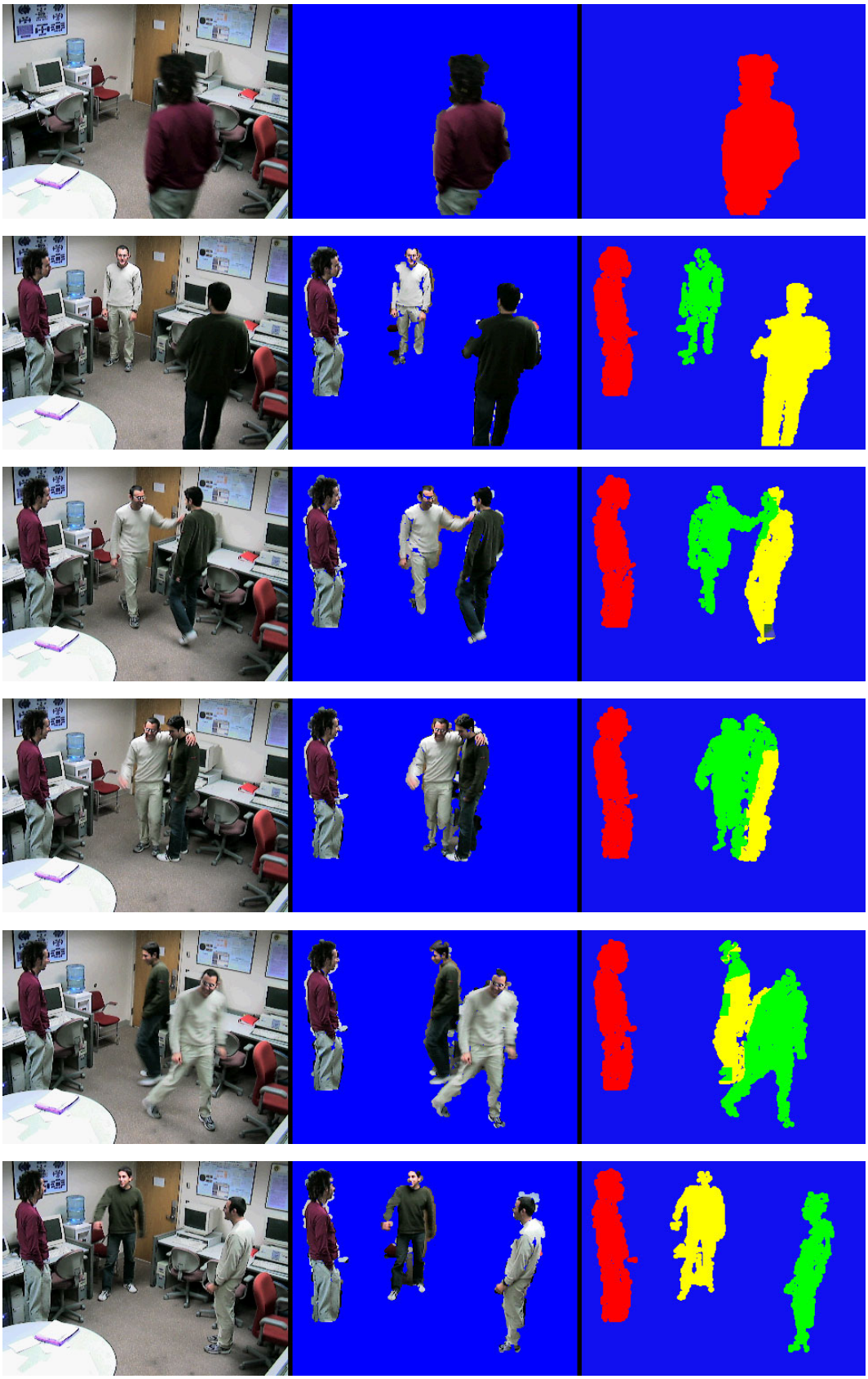


Figure 4.10: A sequence with three persons. Sample frames: 112, 293, 322, 341, 374 and 408. MAP segmentation method was used.

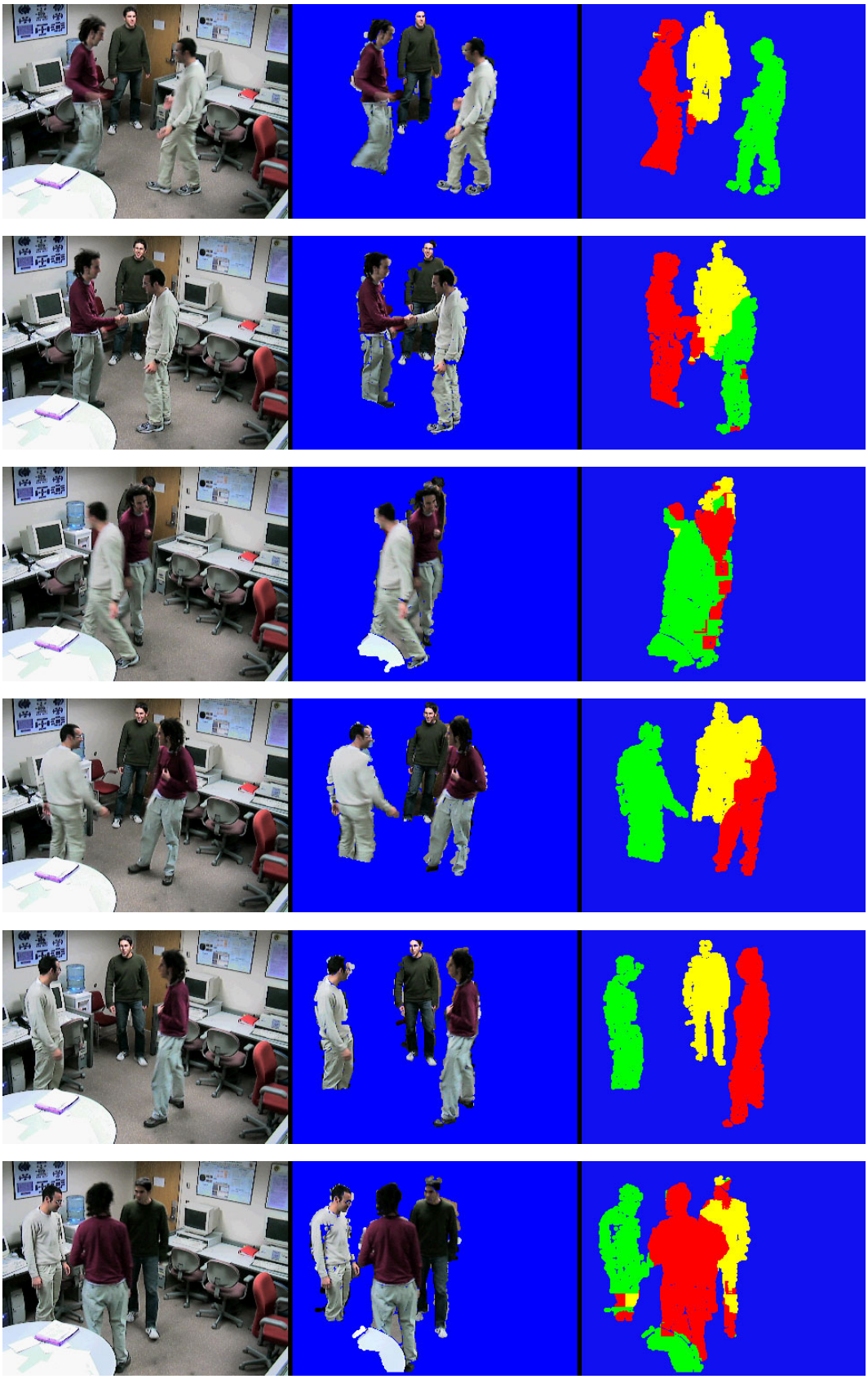


Figure 4.11: A sequence with three persons. Sample frames: 443, 466, 488, 502, 524 and 556. MAP segmentation method was used.

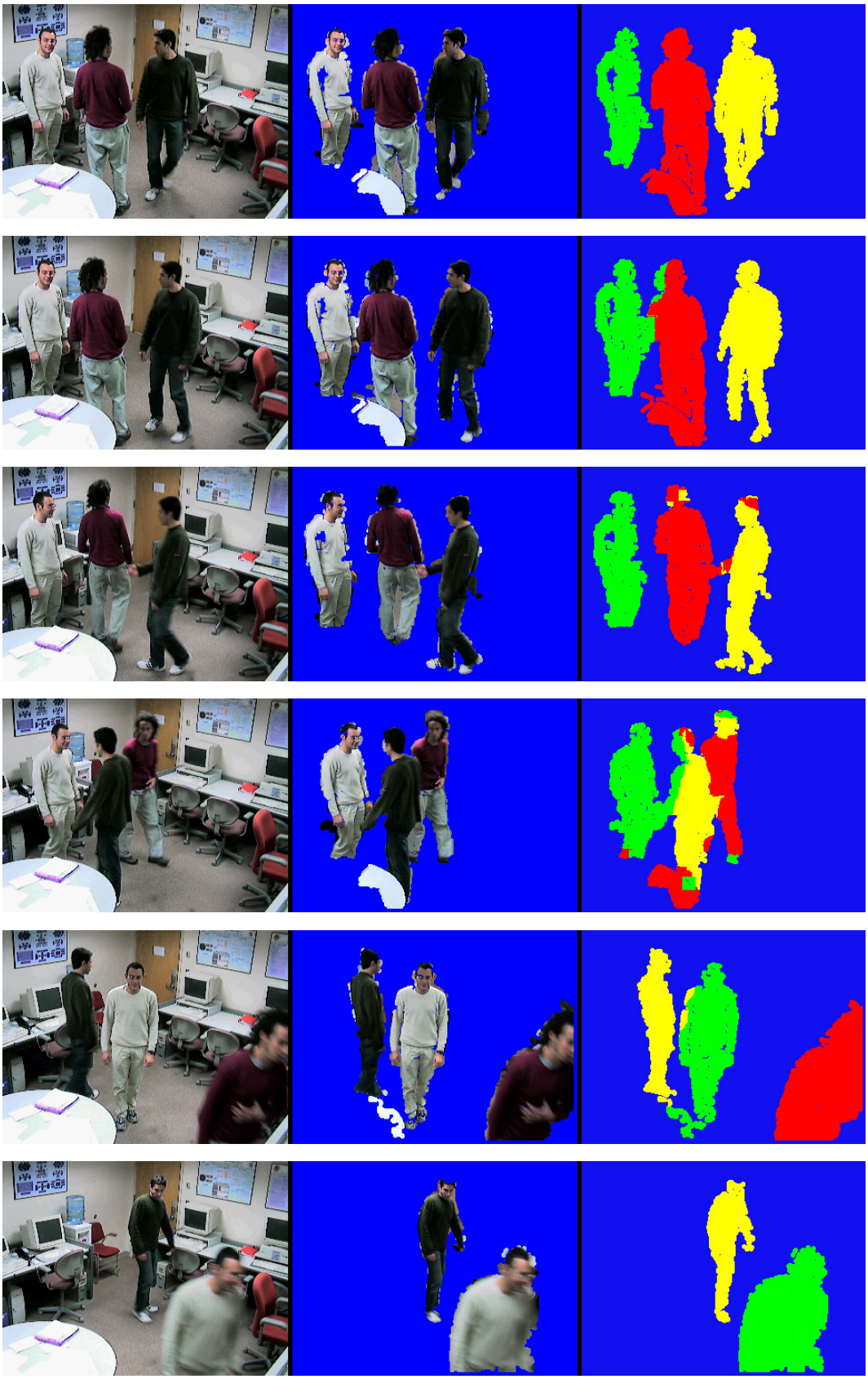


Figure 4.12: A sequence with three persons. Sample frames: 578, 582, 606, 626, 684 and 720. MAP segmentation method was used.

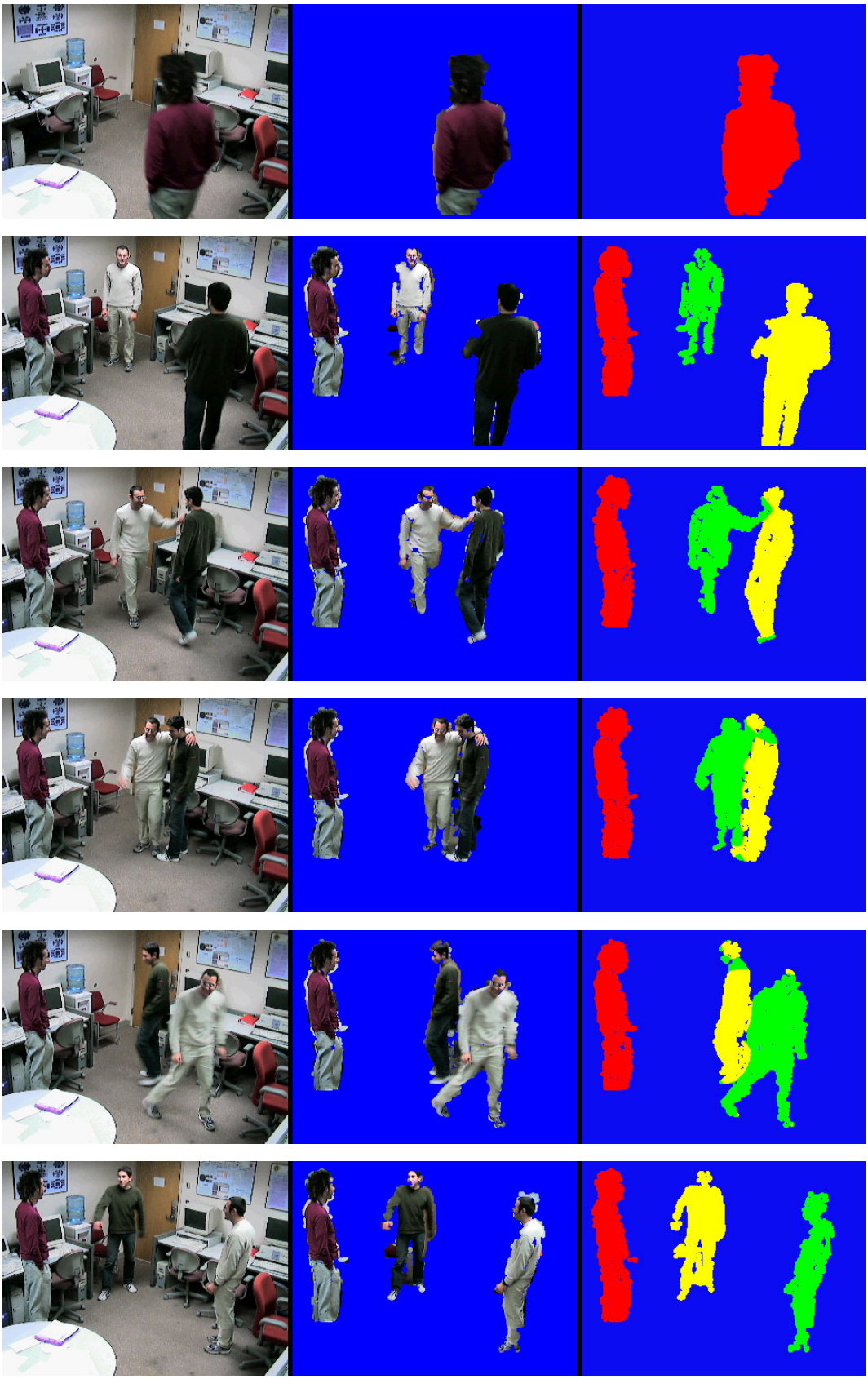


Figure 4.13: A sequence with three persons. Sample frames: 112, 293, 322, 341, 374 and 408. Images quantized to 1000 colors.

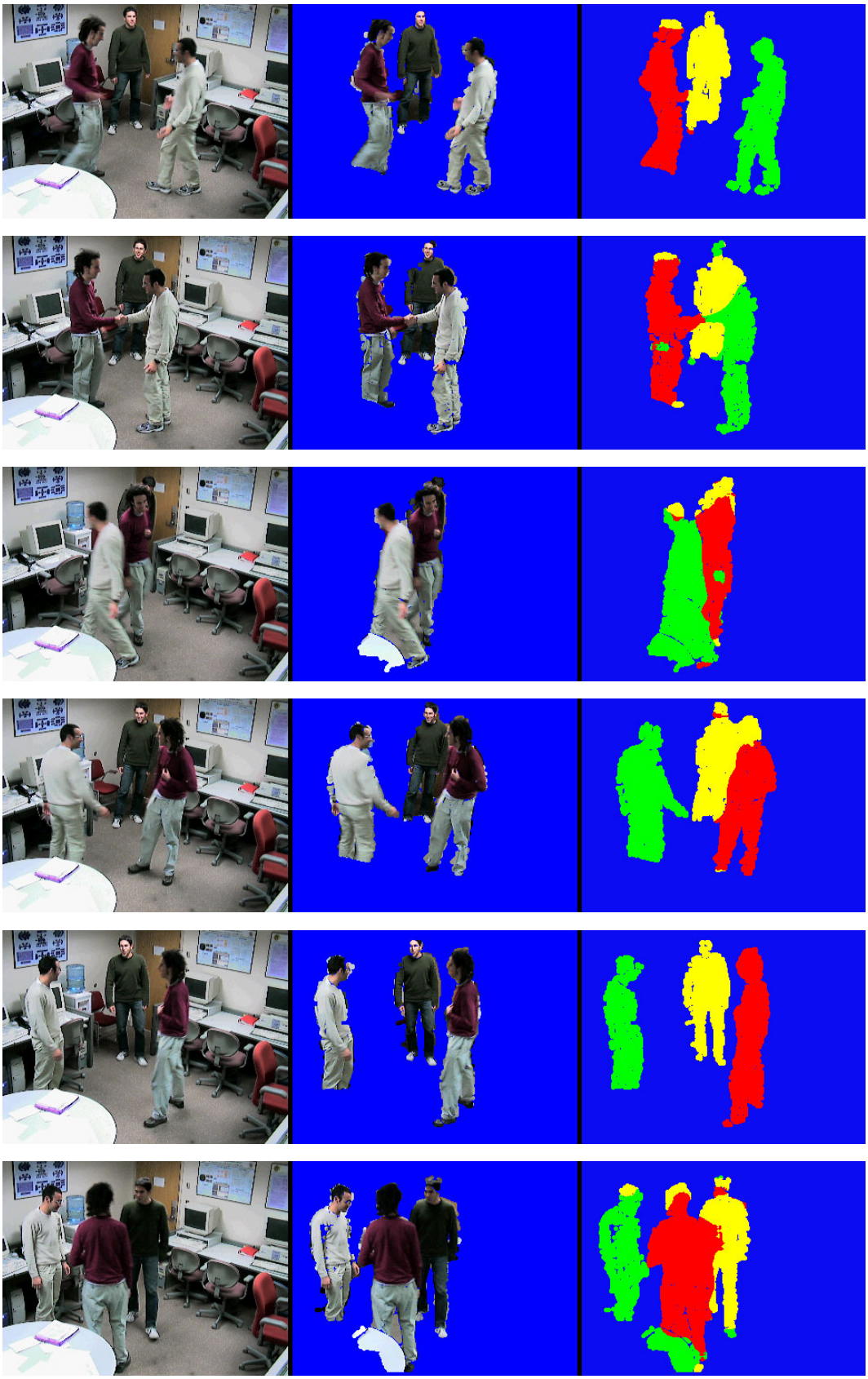


Figure 4.14: A sequence with three persons. Sample frames: 443, 466, 488, 502, 524 and 556. Images quantized to 1000 colors.

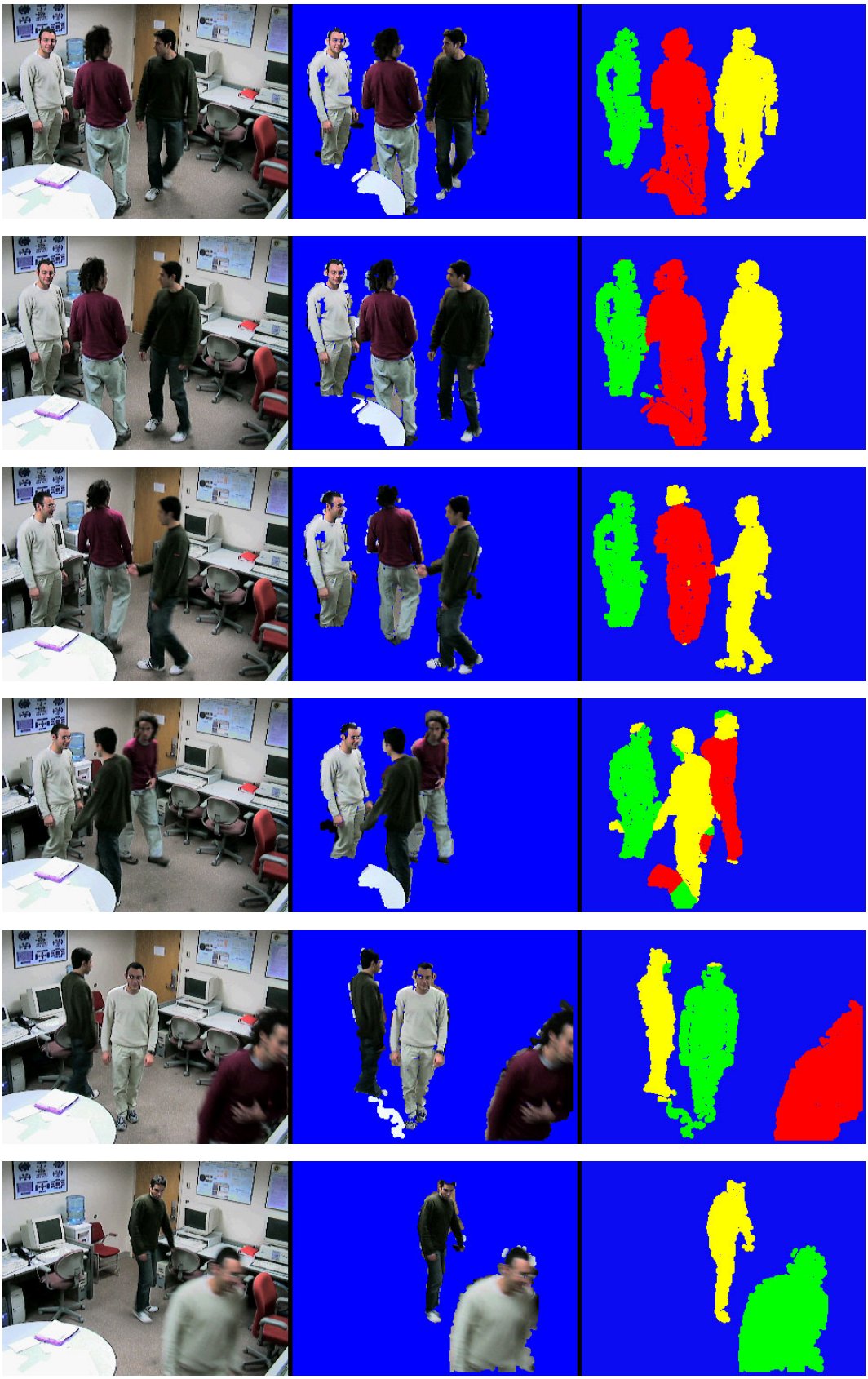


Figure 4.15: A sequence with three persons. Sample frames: 578, 582, 606, 626, 684 and 720. Images quantized to 1000 colors.

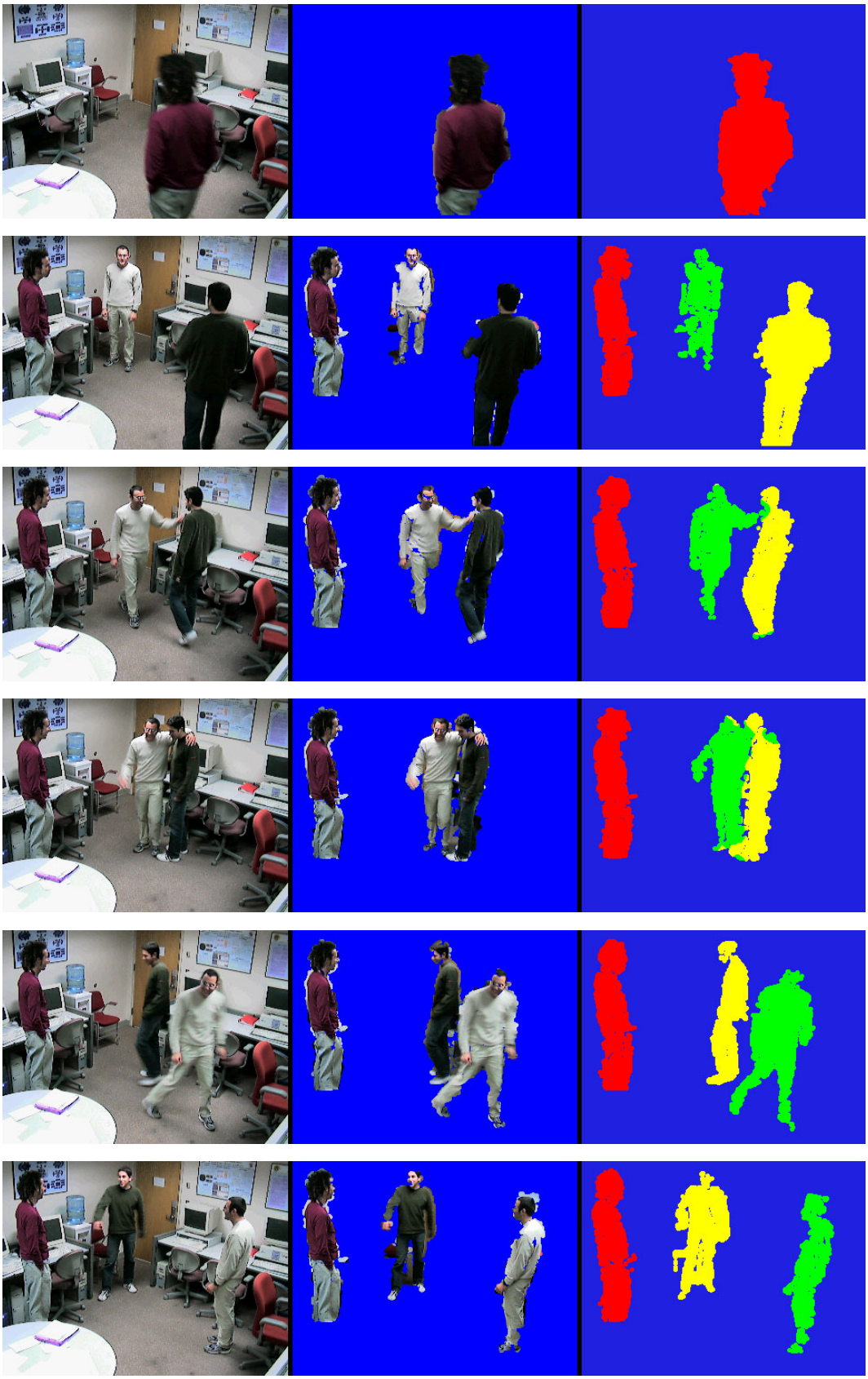


Figure 4.16: A sequence with three persons. Sample frames: 112, 293, 322, 341, 374 and 408. Images quantized to 64 colors.

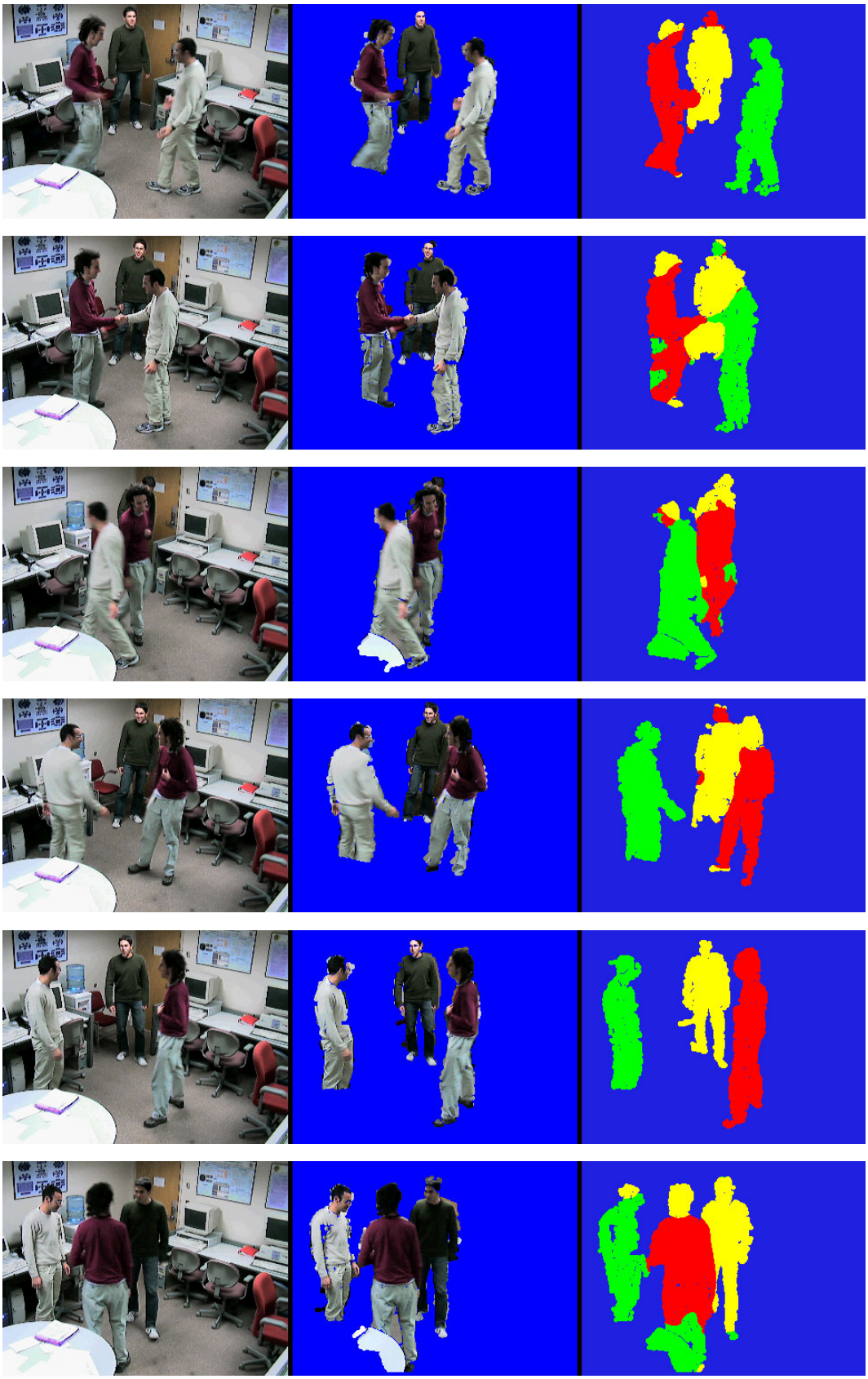


Figure 4.17: A sequence with three persons. Sample frames: 443, 466, 488, 502, 524 and 556. Images quantized to 64 colors.

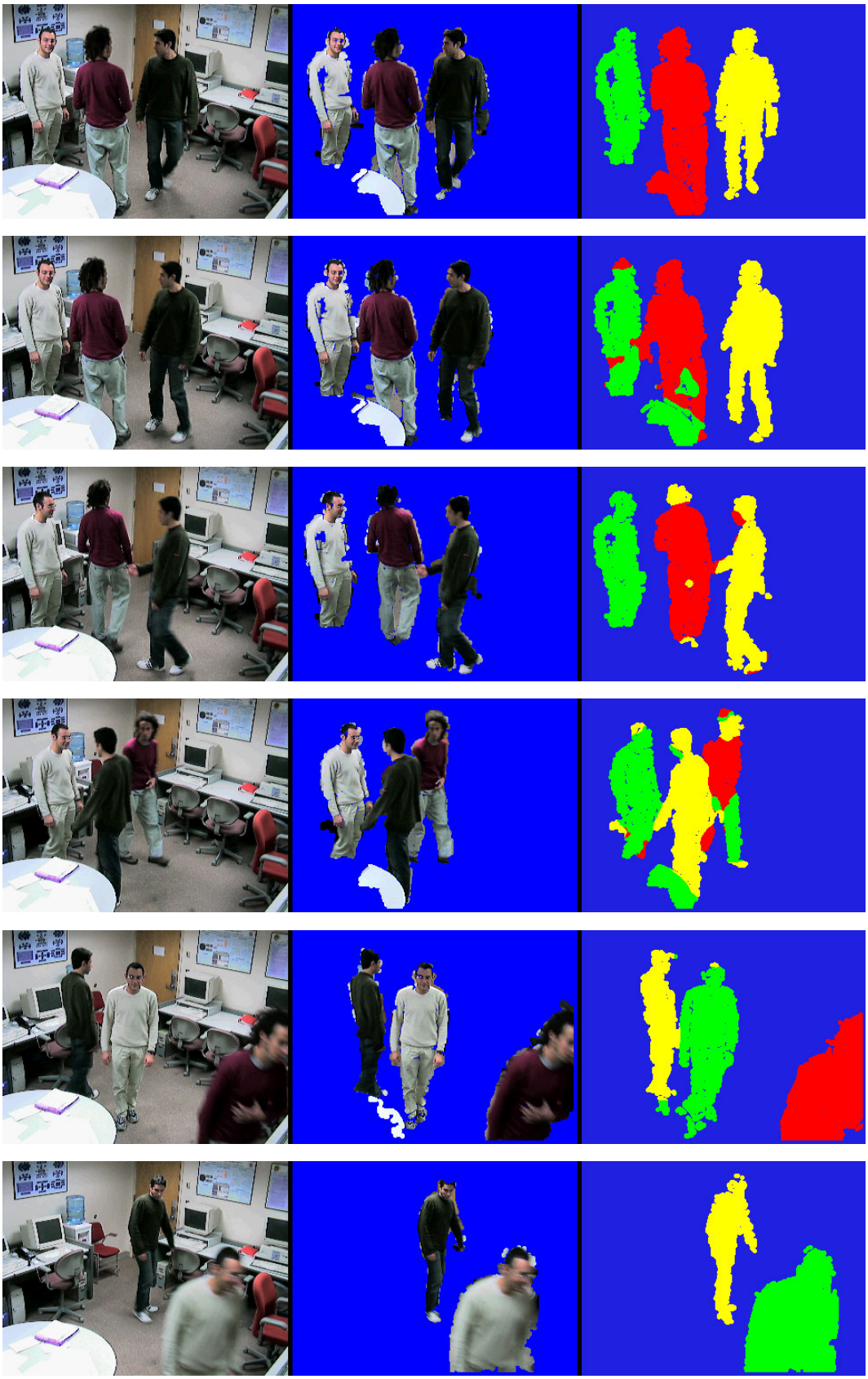


Figure 4.18: A sequence with three persons. Sample frames: 578, 582, 606, 626, 684 and 720. Images quantized to 64 colors.

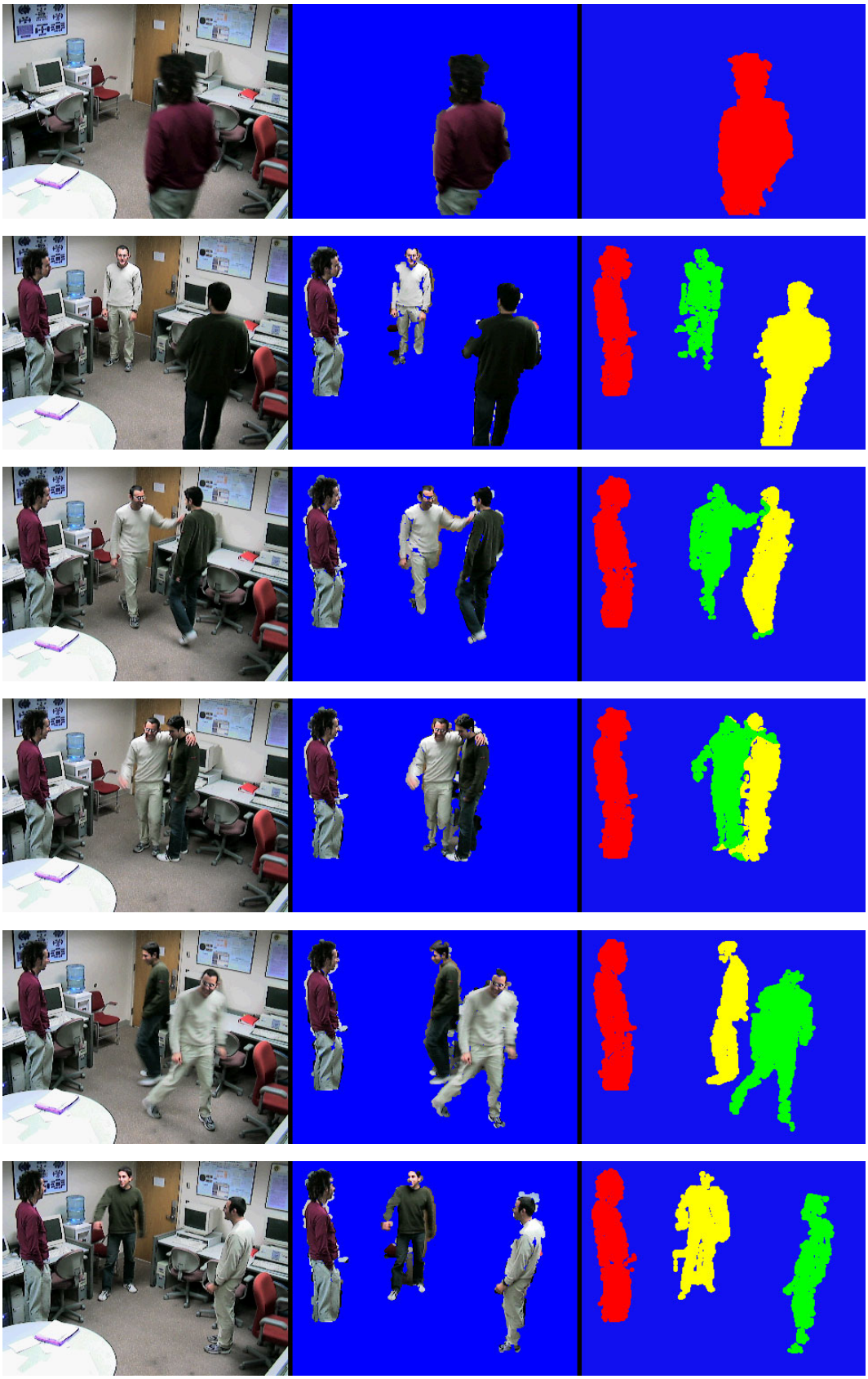


Figure 4.19: A sequence with three persons. Sample frames: 112, 293, 322, 341, 374 and 408. Correlogram calculated at distances $\{1, 3, 5, 7\}$.

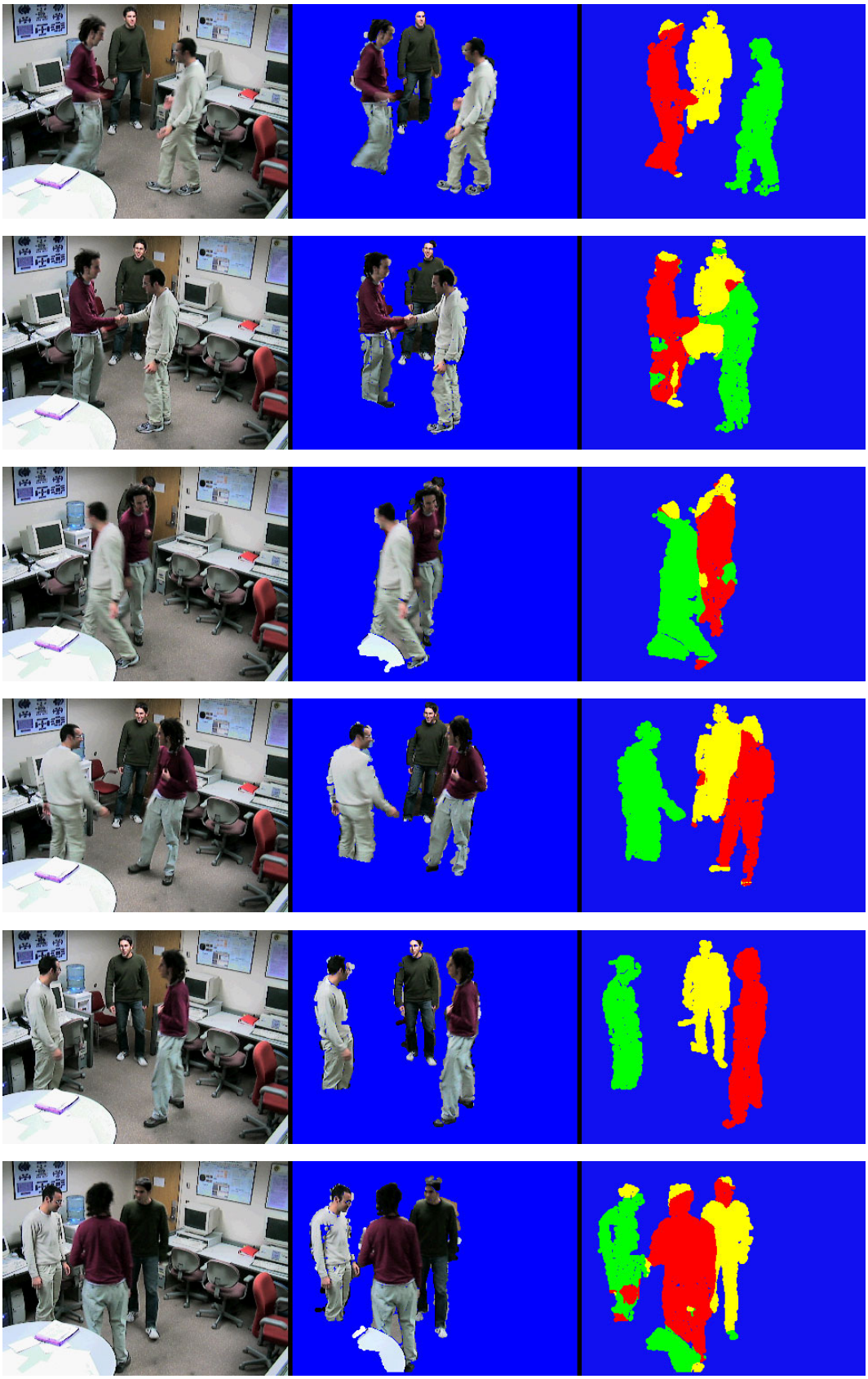


Figure 4.20: A sequence with three persons. Sample frames: 443, 466, 488, 502, 524 and 556. Correlogram calculated at distances $\{1, 3, 5, 7\}$.

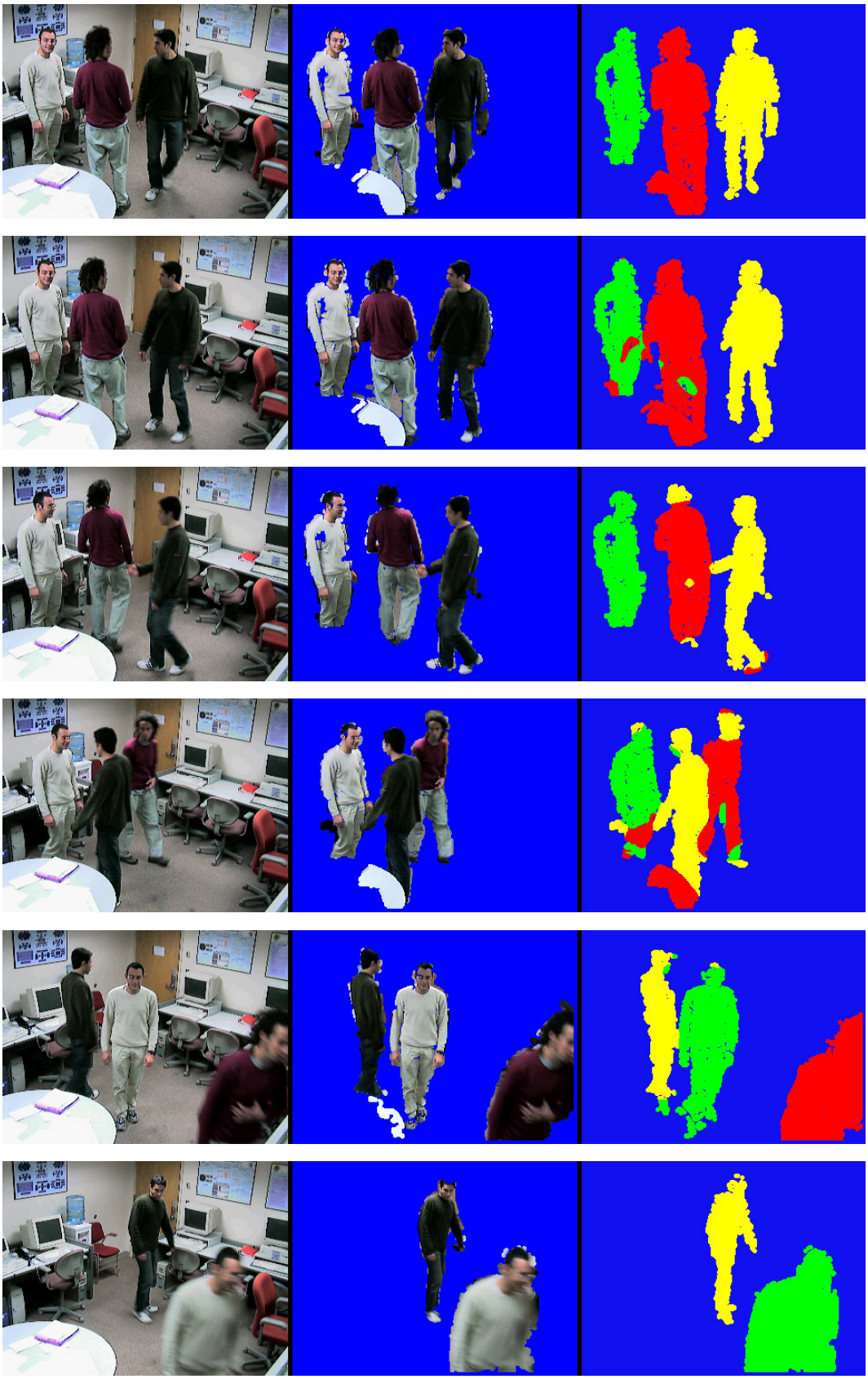


Figure 4.21: A sequence with three persons. Sample frames: 578, 582, 606, 626, 684 and 720. Correlogram calculated at distances $\{1, 3, 5, 7\}$.

people form a group, shadows make the segmentation even more difficult.

4.3.3 Testing correlogram versus histogram

In human tracking, object recognition and image retrieval algorithms [9, 22, 42, 37] histogram has been use to model appearance. However the histogram does not contain any spatial information and that is the main motivation for using the correlogram instead of the histogram. In order to evaluate the increase in the performance due to correlogram, the segmentation was tested in a sequence where there are two people wearing very similar clothes. In Figure 4.25 the segmentation results using only histogram information are shown. Figure 4.26 shows the segmentation results using correlogram as appearance based model. The segmentation method used was the histogram backprojection method with correlogram correction explained in Section 3.5.1. Clearly the correlogram segmentation in Figure 4.26 is much better than the histogram segmentation shown in 4.25.

4.3.4 Testing the performance for compressed data

The performance of the system for compressed data has also been tested. The sequence was compressed using MPEG1 at a 700Kbps rate. The segmentation results are shown in Figure 4.27. It can be seen that the performance decreases but the algorithm is still able to track de people. Probably reducing the number of colors in the quantization process would help to reduce the effects of the MPEG noise, while not hurting too much the quality of the segmentation. It turns out that the background subtraction algorithm is also affected by the MPEG noise and therefore, some of the decrease in the quality of segmentation is due to the decrease in the quality of the output of the background subtraction algorithm.

4.3.5 Testing identification performance

In this experiment the human identification performance is tested. The test is done over a small data base of five humans, entering the scene in a random order. At the end, each individual has been in the room twice. The first time a person enters the scene, a model is built. After, each time a person enters the scene, the algorithm measures the distance (see (3.7)) between the observed person and the stored models. If the distance is above certain threshold that is set empirically (usually 0.55 works fine), it is concluded that the person had not been modeled before, and a new model is initialized on the fly. Otherwise it is matched with the closest model. In this experiment all individuals were properly identified. Sample frames are shown in Figures 4.28 and 4.29. For this experiment the images were quantized to 512 colors. In general, the number of colors needed in order to get good results for identification is less than the number of colors needed for segmentation. The distance set where the correlogram was calculated is: $\{1, 3, 5, 7, 10, 15, 23, 35\}$. A smaller distance set may be sufficient for achieving the same results.

It is not claimed that the algorithm will always achieve a 100% detection rate with a 0% false alarm rate. The performance will depend on clothing, and how close they are to each other. No

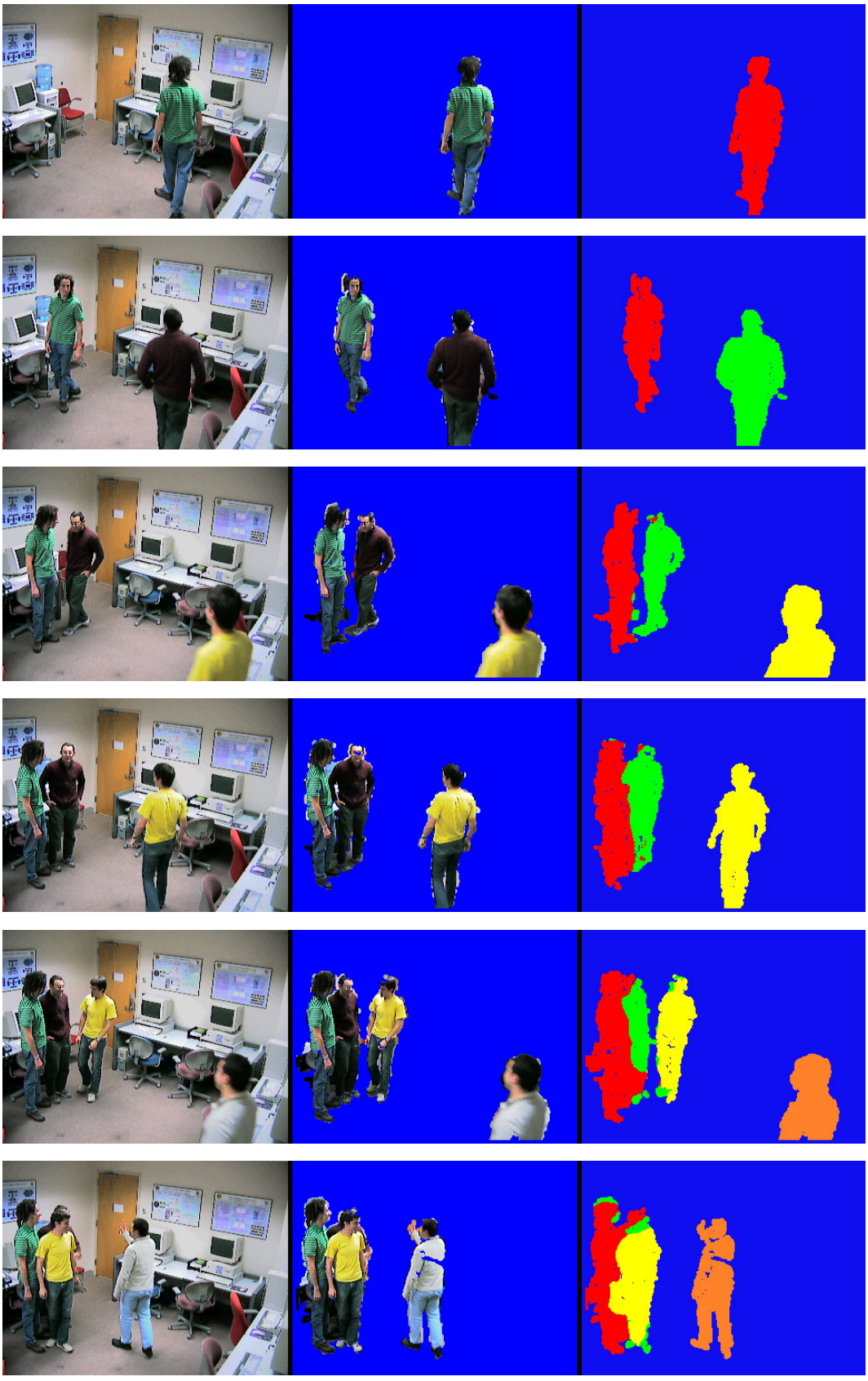


Figure 4.22: Sequence with five persons. Sample frames: 126, 236, 293, 346, 425 and 508.

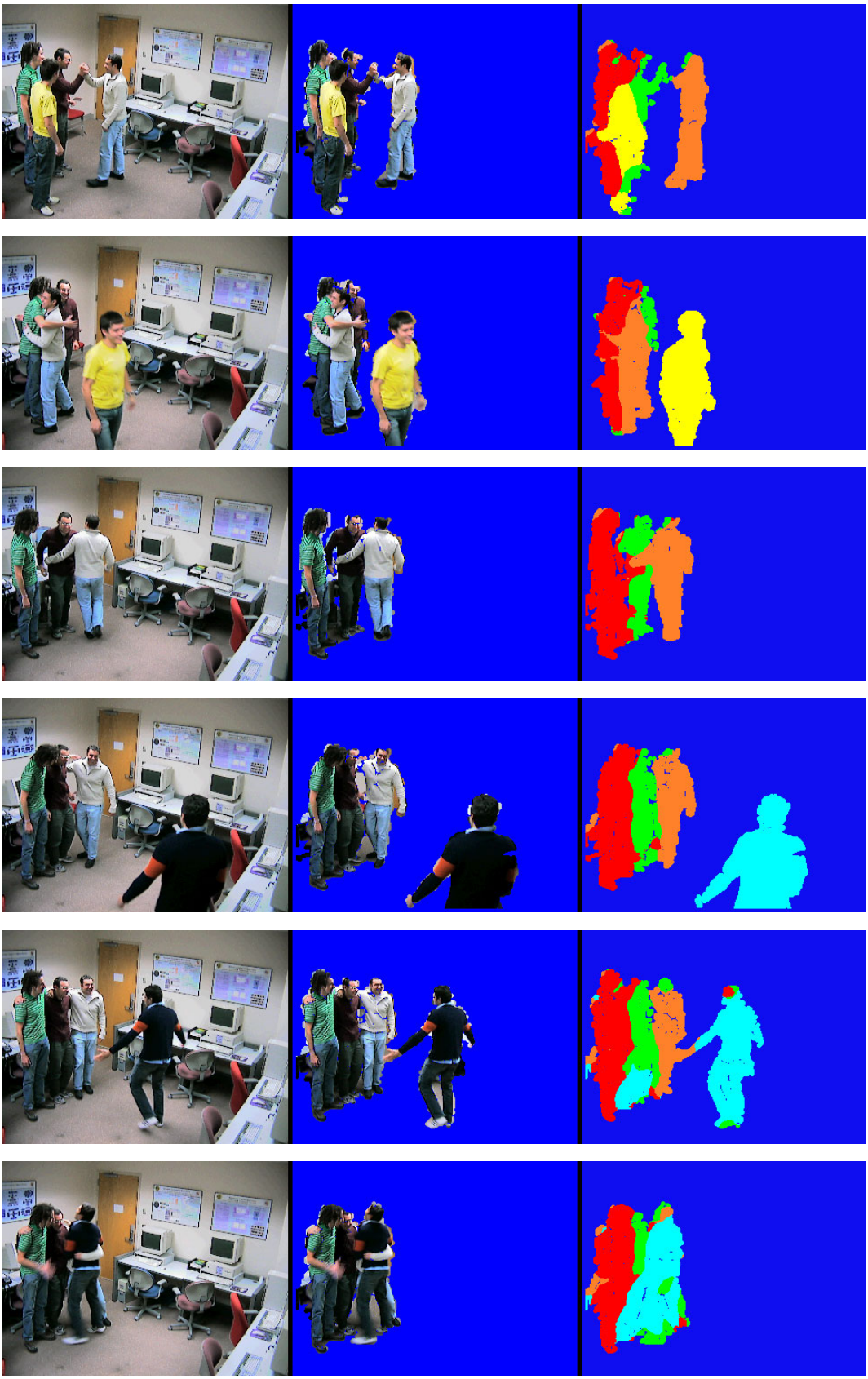


Figure 4.23: Sequence with five persons. Sample frames: 543, 597, 681, 711, 762 and 809.

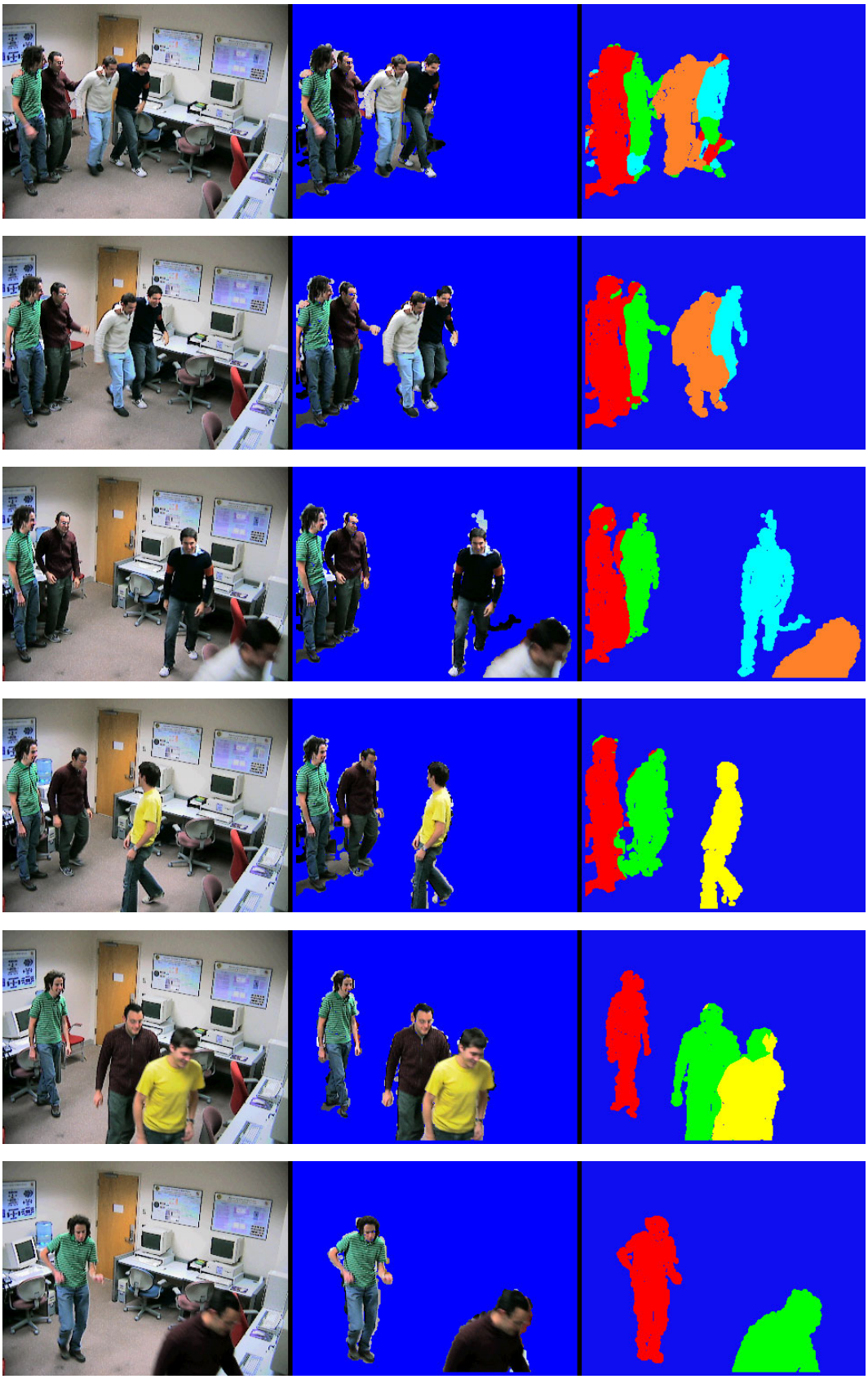


Figure 4.24: Sequence with five persons. Sample frames: 895, 908, 988, 1165, 1230 and 1265.

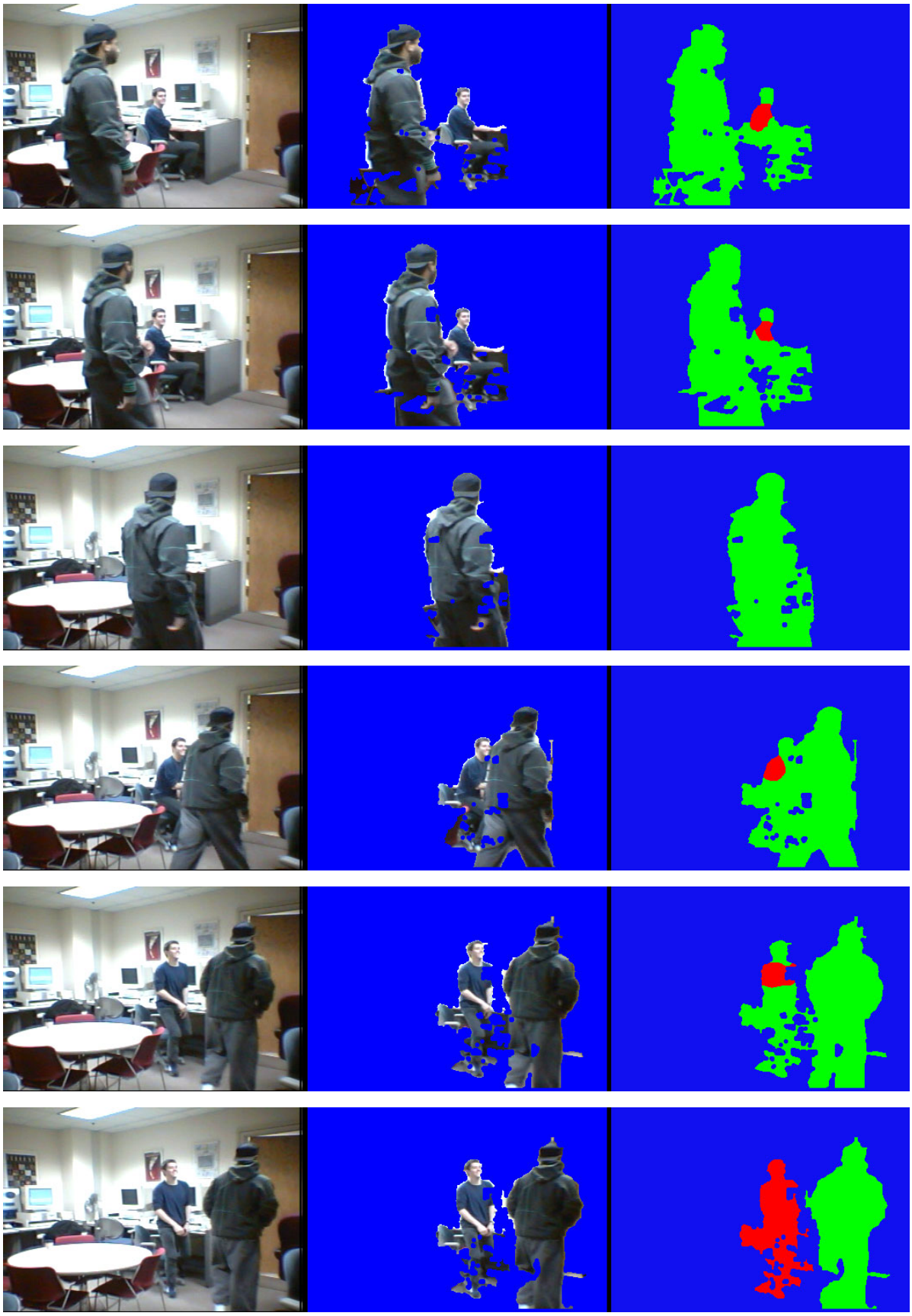


Figure 4.25: Segmentation using the histogram. Sample Frames 467, 470, 478, 508, 515, 517.

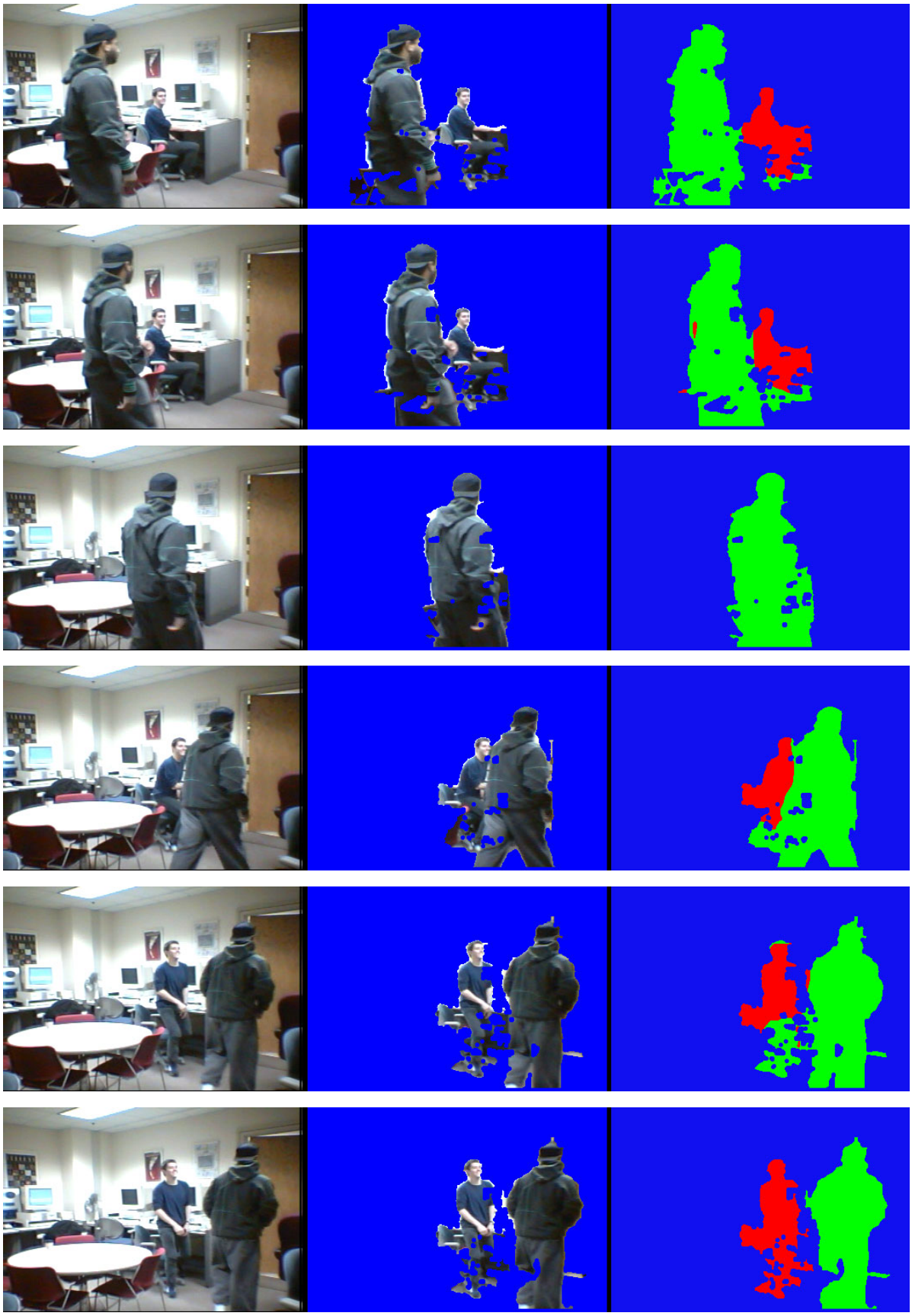


Figure 4.26: Segmentation using the correlogram. Sample Frames 467, 470, 478, 508, 515, 517.

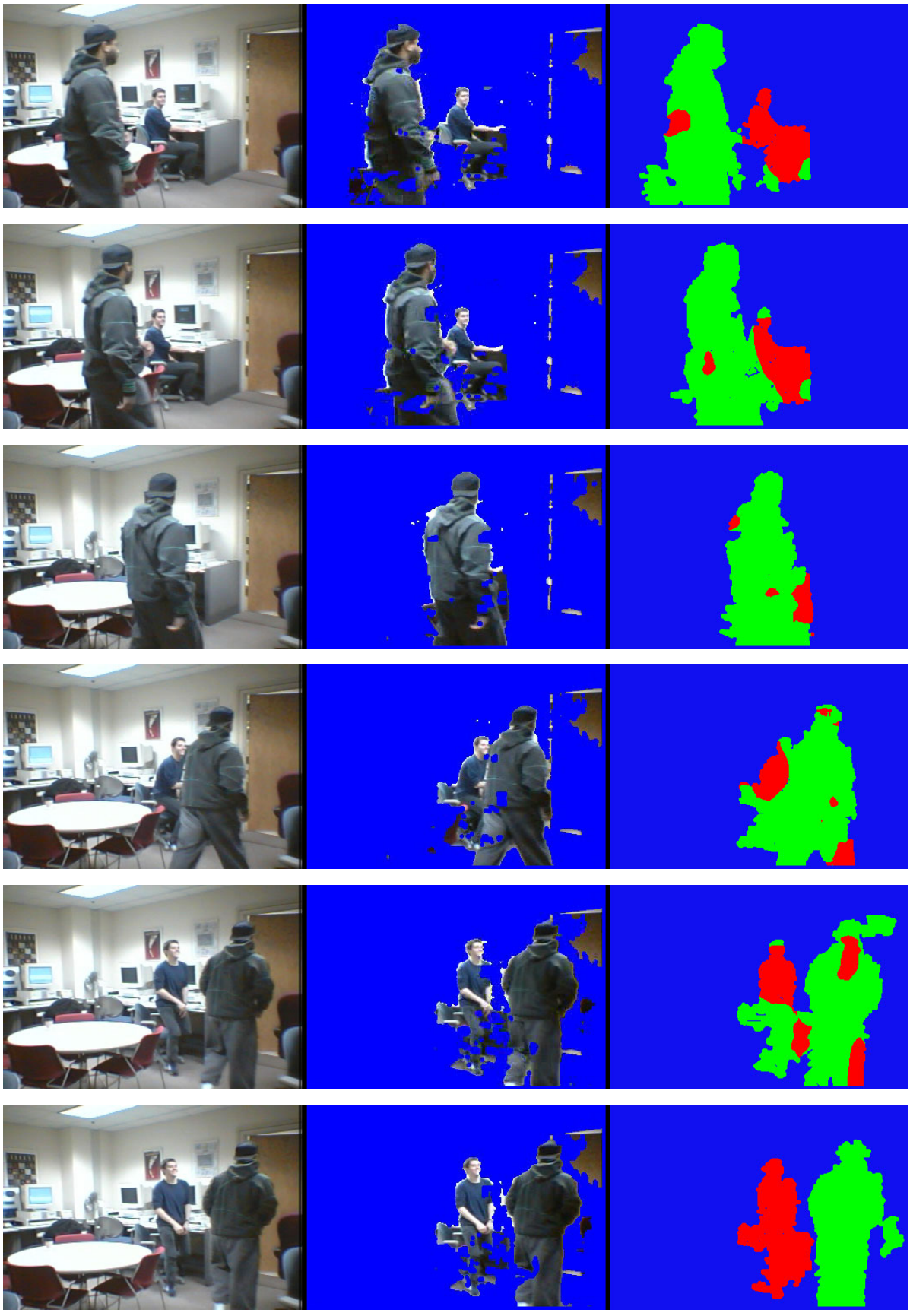


Figure 4.27: Segmentation for a 700Kbps MPEG sequence. Sample Frames 467, 470, 478, 508, 515, 517.

experiment has been done in order to test how many different people could the system handle on an average, or how the detection and false alarm rates would vary when changing the thresholds on the similarity measure. If it is assumed that all the persons in the scene have been previously modeled, then the system would be able to handle a larger number of people.

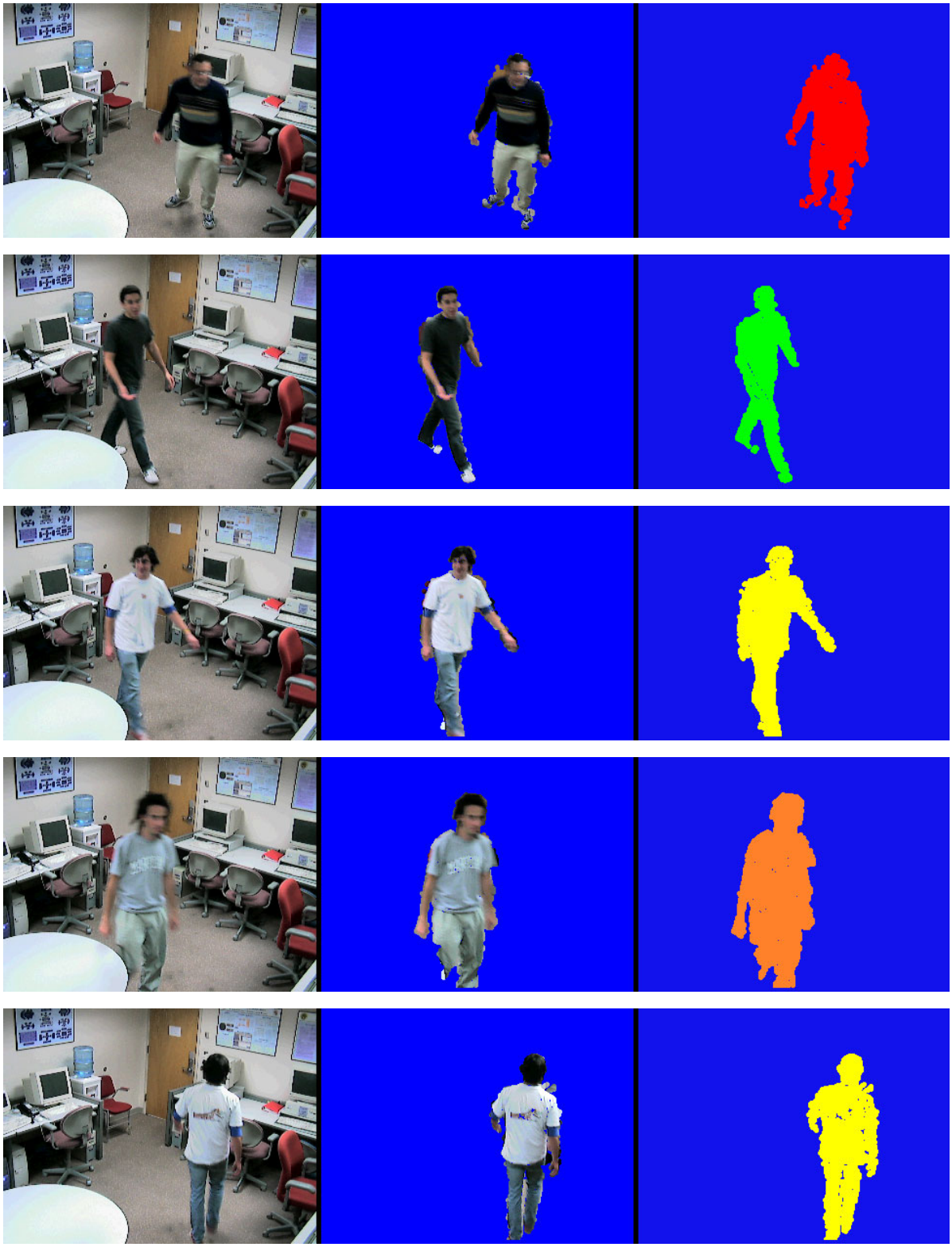


Figure 4.28: Identification test. Sample frames 93, 149, 210, 290 and 404.

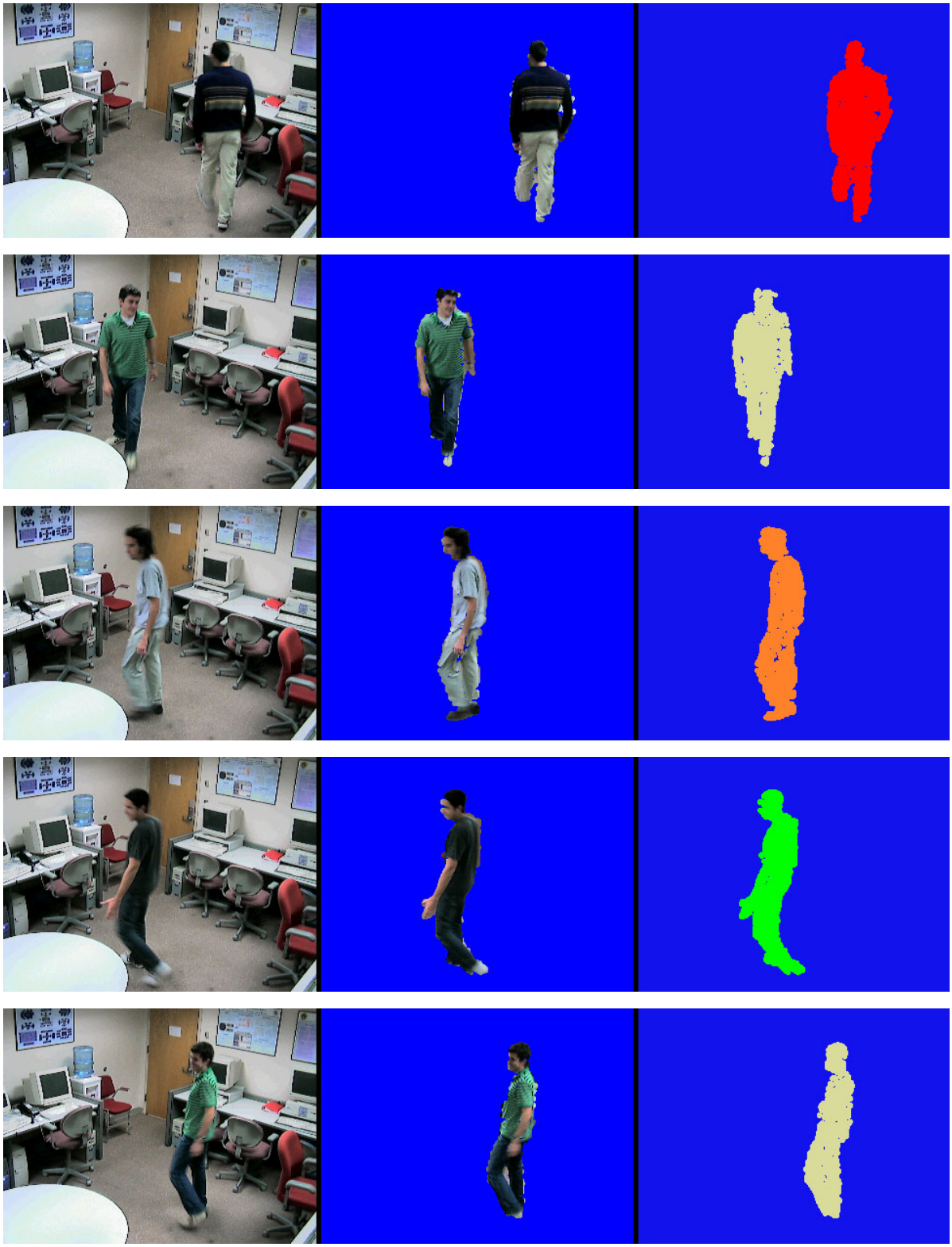


Figure 4.29: Identification test. Sample frames 541, 662, 768, 872 and 952.

Chapter 5

Interpretation of human-object interactions

5.1 Introduction

Detecting interactions between human and objects is of fundamental importance in automatically inferring human activities. A video, for example, could be summarized by showing only the frames were people are interacting with objects. In surveillance applications it is obviously useful to determine when someone is interacting with objects (to know, for instance, if anybody is stealing anything, or to know if anybody has left a suspicious object).

To our knowledge, there has been very little work reported on this problem. In [7] Haritaoglu et al. were able to segment objects that were carried by people using a simple symmetry constraint on the shape of the people. In the system that McKenna et al. implemented [9], they detected the event of removing or depositing an object in the scene, but they did not differentiate between them and they did not try to track the objects.

In the next section a system that is able to detect when someone has removed or deposited an object in the scene is presented. Once such an event is detected the system builds a model for the person and the object. In the case of an object being removed the system uses the models to track the object and the person for the rest of the sequence. If the object is deposited, the system goes back and tracks the object and the person retroactively. In Figure 5.1 a flowchart of the algorithm is shown.

5.2 The algorithm

Like in the human tracking system, the system for detecting human-object interactions is based on the background subtraction algorithm. When a person deposits an object in the scene, since the object has not been modeled in the learning stage of the background subtraction algorithm, it will be detected as a foreground region. The same will happen if a person takes an object from the scene. Since part of the scene that was behind the object was not modeled, when the object

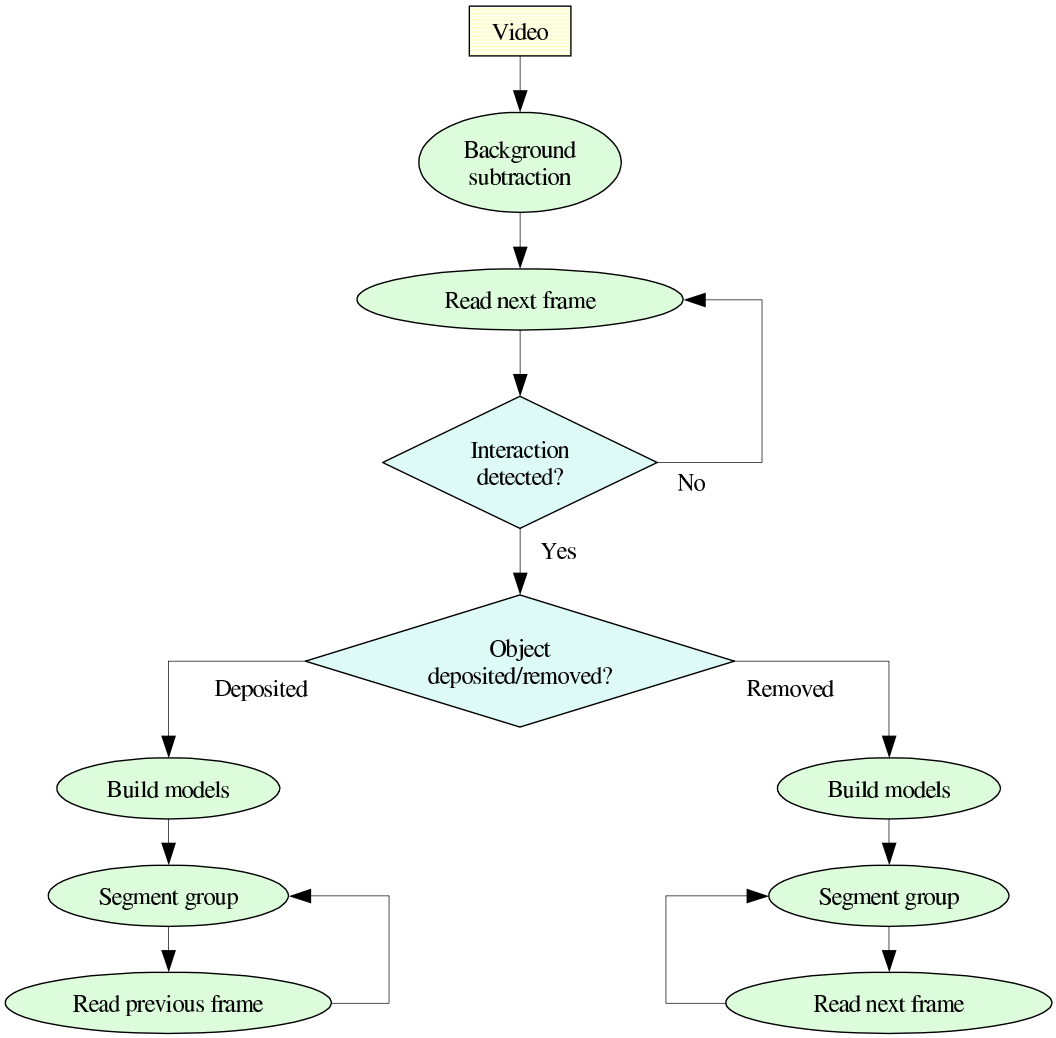


Figure 5.1: The flowchart of object tracking algorithm diagram.

is removed a new foreground region will appear. Therefore, the first objective of the algorithm is detecting such events. The system will detect that a new object has been deposited in the scene or that an object has been removed when it detects that a blob splits into two blobs and one of them is static. Static means that the centroid of the blob does not move more than 2 pixels from it initial position during a certain period of time (for example 10 frames). In images (a) and (b) of Figures 5.4 and 5.3, an example of an object being deposited and removed respectively can be seen.

Figure 5.2: Background subtraction mask showing a human-object interaction. From the mask it cannot be decided whether the object has been deposited or removed. Although it may seem the opposite, in this case the object has been removed.

From the background subtraction alone however, it cannot be determined whether the object has been removed or deposited in the scene, since in both cases the foreground masks would look exactly the same. This can be observed in Figure 5.2. Although it may seem that an object has been deposited, actually it is the opposite case. Therefore, in order to differentiate the event of taking an object from the event of depositing an object, the images have to be analyzed more carefully. Let us assume that the position of the pixels that lie in the boundaries of the object are known. Then, if the gradient response is calculated in these pixels it will be possible to know if the object is present or not. When the object is present, the gradient response will be higher, since the object edges will be present. The system can know the position of the pixels lying in the boundaries of the object using the mask of the background subtraction algorithm. Therefore the system can differentiate between an object being picked up and object being deposited. Mathematically these concepts can be expressed in the following way:

Let $\nabla(p)$ be the gradient response at pixel p. Let the Sobel masks in each direction be:

$$\nabla_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \qquad \nabla_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \tag{5.1}$$

then,

$$\nabla(p) = \sqrt{\nabla_x^2(p) + \nabla_y^2(p)} \tag{5.2}$$

Let B be the set of pixels lying in the boundary of the object, and let $B_i = (x, y)$ be the i^{th} element of this set. Let $\mathcal{I}_B$ be the background image and $\mathcal{I}_D$ be the frame where the interaction has been detected. Let

$$r_B = \sum_{\forall p \in B} \nabla(\mathcal{I}_B(B_p)) \tag{5.3}$$

and

$$r_D = \sum_{\forall p \in B} \nabla(\mathcal{I}_D(B_p)) \tag{5.4}$$

Then, if $r_B > r_D$ the system will decide that the object has been taken. Otherwise, the system will decide that the object has been deposited. In other words, if in the background frame the gradient response along the object boundary is greater than in the frame where the interaction has been detected, it is decided that the object has been removed (see Figure 5.3). In the other case, if the gradient response along the object boundary is greater in the frame where the interaction has been detected than in the background frame, the object has been deposited (see Figure 5.4).

After a human-object interaction is detected, if the object has been removed, the system builds a model for the person and a model the object. This model is initialized when the object and the person are isolated in the image plane (when there is no occlusion between them) and then, using these models and the method described in Section 3.5.1 the object and the person are segmented for the rest of the sequence. To minimize the effects due to changes in illumination conditions, the same normalization method described in Section 4.2 is used. If, on the other hand, a new object has been deposited, the models are used to segment the person and the object back in time, from the point when the person entered the scene until the moment when the object was detected.

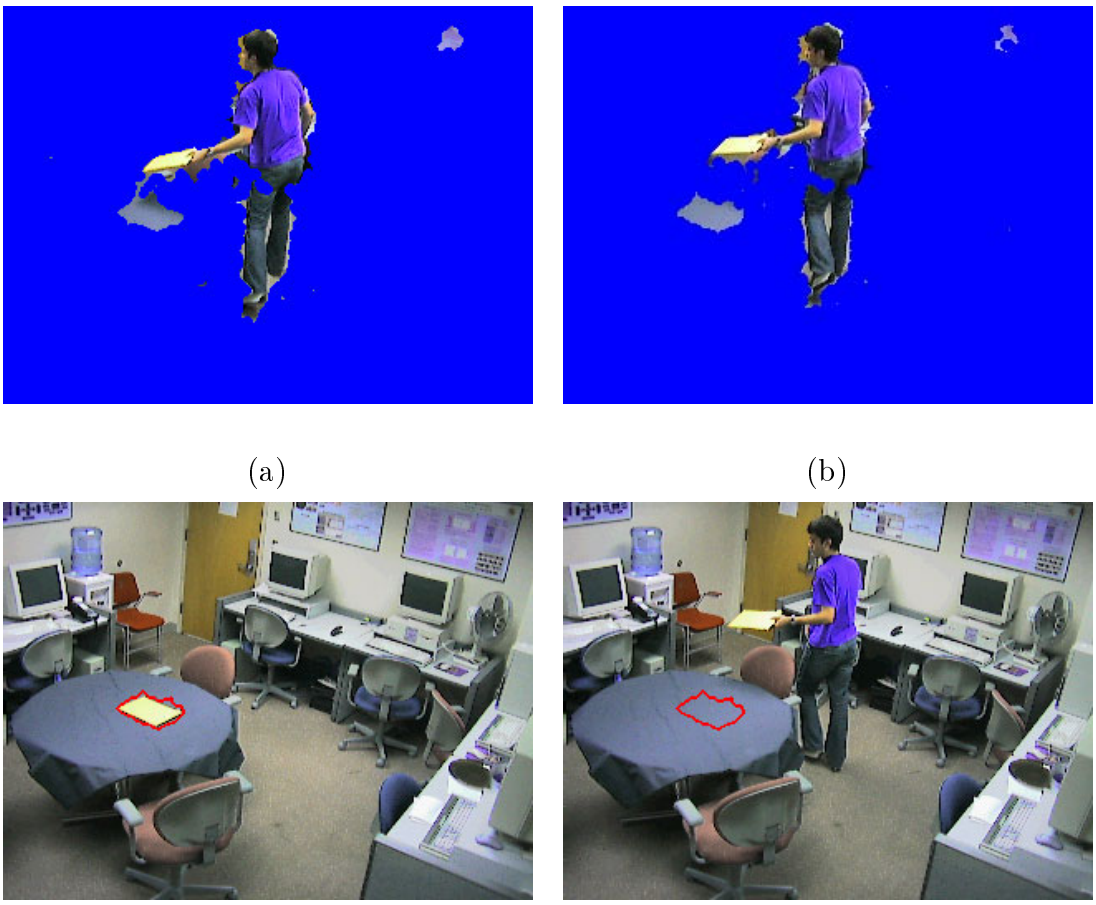
5.3 Results

Several sequences have been recorded in order to test the algorithm for detecting interactions of human with objects. The objective of the experiments is to show that the algorithm can distinguish between an object being removed and an object being deposited, and that it can track different kinds of objects. In all the experiments done, the system was able to differentitate between and object being picked up and an object being deposited.

In all experiments the correlogram was calculated at distances $\{3, 6, 9, 12, 15, 20\}$ with the images quantized to 512 colors.

5.3.1 Testing an object being removed

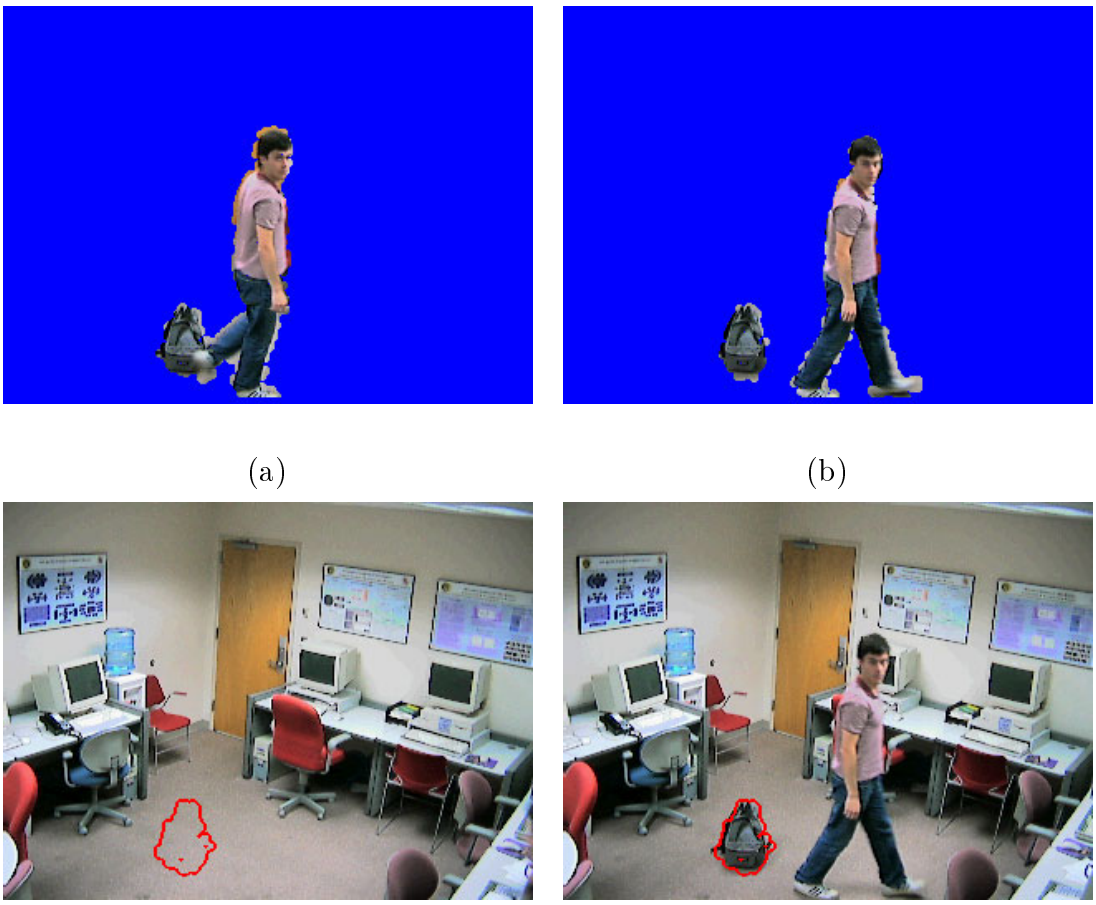
The algorithm has been tested with two different objects, a folder and a bag. The folder is easier to track since its color is very different from the colors of the clothes of the person. The results can be seen in Figures 5.5, were the segmentation is almost perfect. For the other sequence the colors are much more challenging and the illumination changes at some points of the clip introduce error in segmentation, as can be seen in Figures 5.6 and 5.7.



(a) (b)

(c) (d)

Figure 5.3: Example of an object being picked up. In (a) and (b) the background subtraction output before and after the object has been removed is shown. In (c) and (d) the gradient response is calculated along the object boundaries in the background frame (c) and the frame where the interaction has been detected (d). In this case the gradient response is larger in (c) than in (d), therefore it is concluded that the object has been picked up.



(a) (b)

(c) (d)

Figure 5.4: Object depositing example. In (a) and (b) the background subtraction output before and after the object has been deposited is shown. In (c) and (d) the gradient response is calculated along the object boundaries in the background frame (c) and the frame where the interaction has been detected (d). In this case the gradient response will be larger in image (d) than image (c), therefore it is concluded that the object has been deposited.

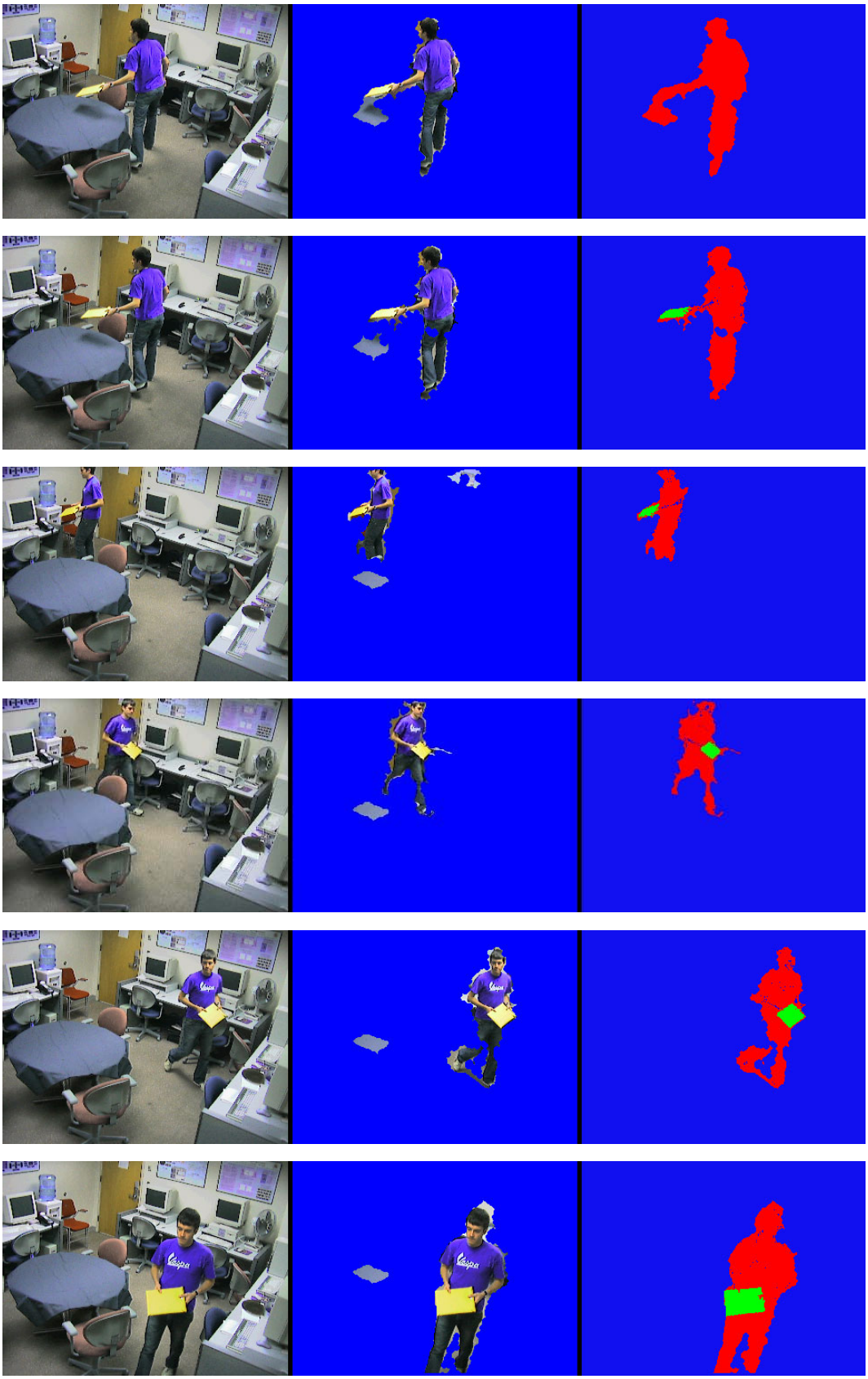


Figure 5.5: Sequence showing a folder being picked up. Sample frames 213, 216, 266, 343, 385, 441.

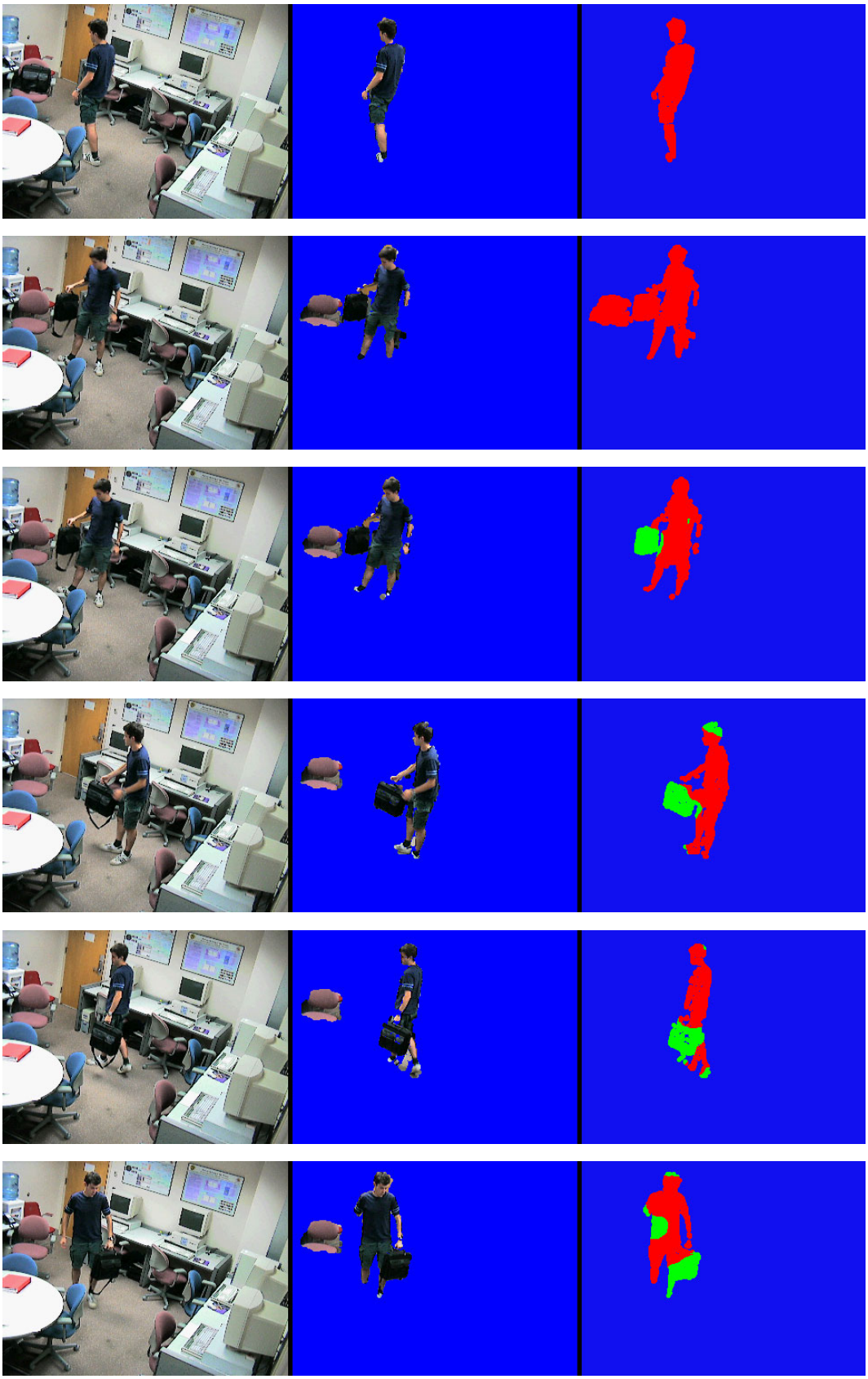


Figure 5.6: Sequence showing a laptop bag being picked up. Sample frames 102, 231, 233, 302, 349, 386.

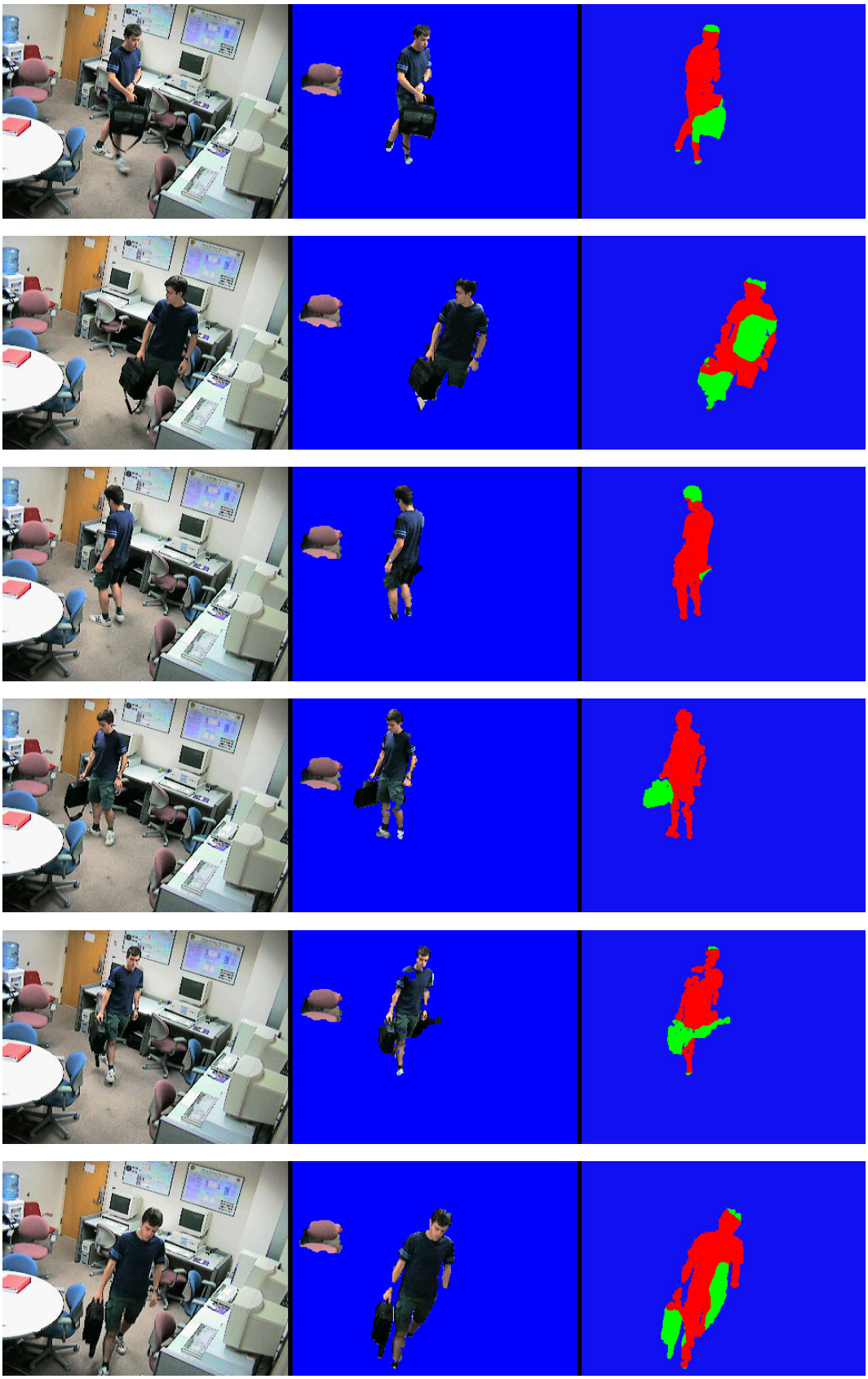


Figure 5.7: Sequence showing a laptop bag being picked up. Sample frames 413, 450, 510, 535, 575, 624.

5.3.2 Testing an object being deposited

The algorithm has been tested in two different sequences. The first sequence shows a person entering a room, taking off a jacket that he is wearing and leaving it on a chair. When the person leaves the jacket on the chair, at frame 845 (see third image in 5.8), the algorithm detects the event, initializes the color models and goes back in time to track the person and the jacket. It is interesting to point out that the algorithm is able to track a deformable object as long as the view of the object when it is initialized is representative enough. In Figures 5.8 and 5.9, it can be seen that the performance is fairly good. Some regions are missclassified(for example the head of the person) due to changes on illumination.

The other experiment shows a person entering to the scene and leaving a bag on the floor. As the bag is occluded by the person, the system detects the event at frame number 664, when the bag becomes visible (see second and third pictures in Figure 5.10). Remaining images in Figures 5.10 and 5.11 show the system going back in time and segmenting the person and the bag, using the models initialized in frame 664.

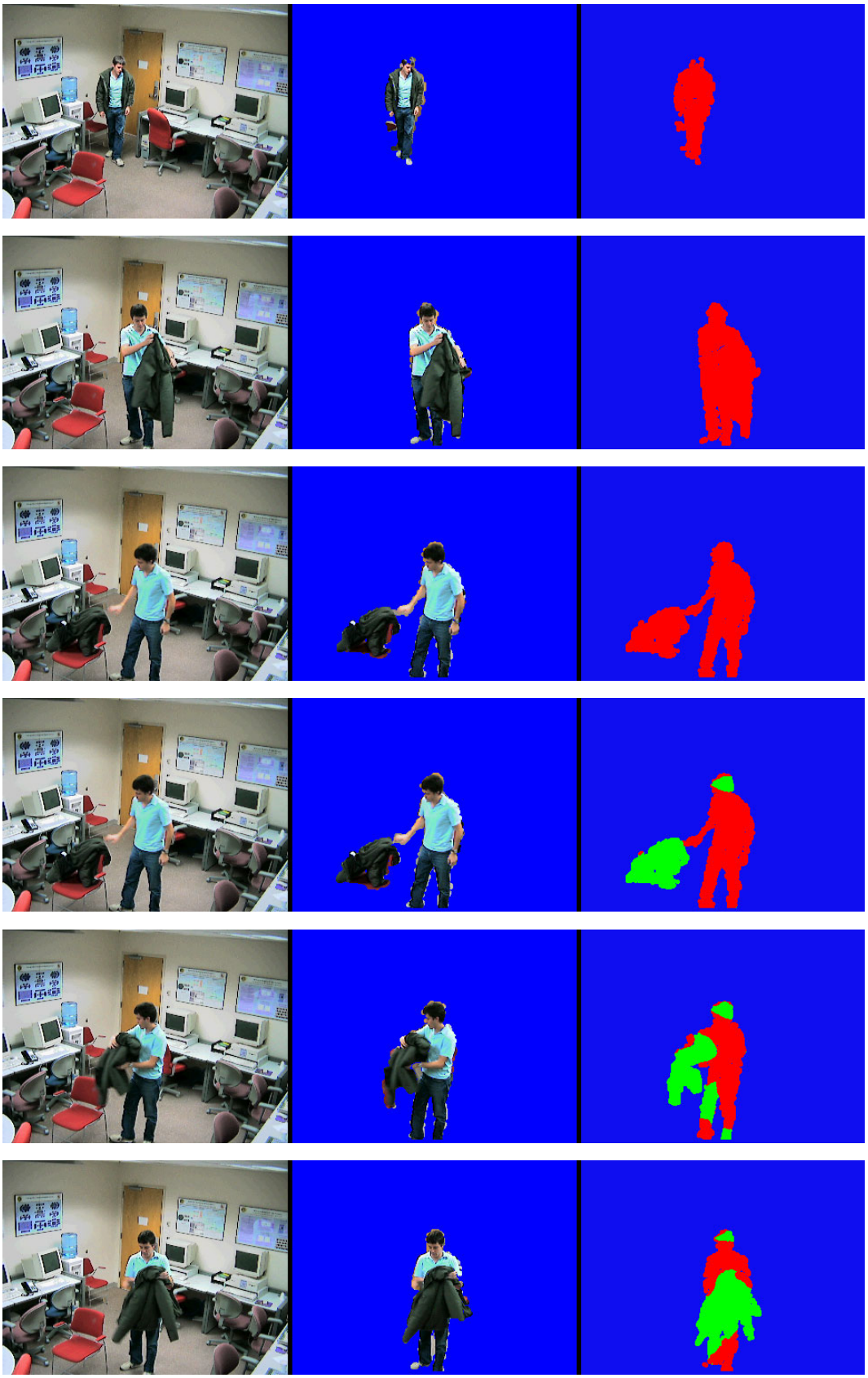


Figure 5.8: Sequence showing a jacket being deposited on a chair. Sample frames 455, 714, 845, 845, 775 and 731

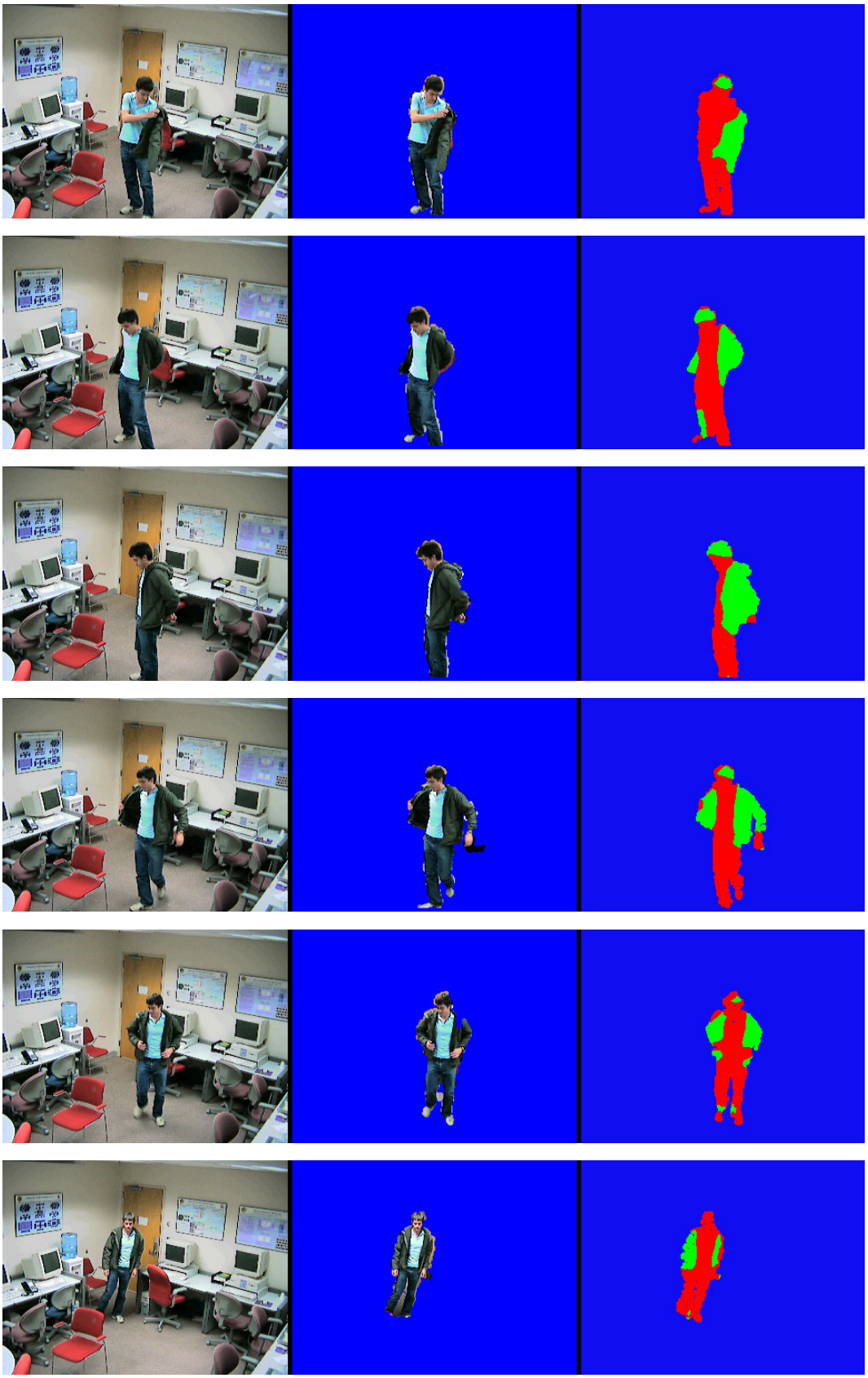


Figure 5.9: Sequence showing a jacket being deposited on a chair. Sample frames 688, 595, 567, 531, 514 and 472.

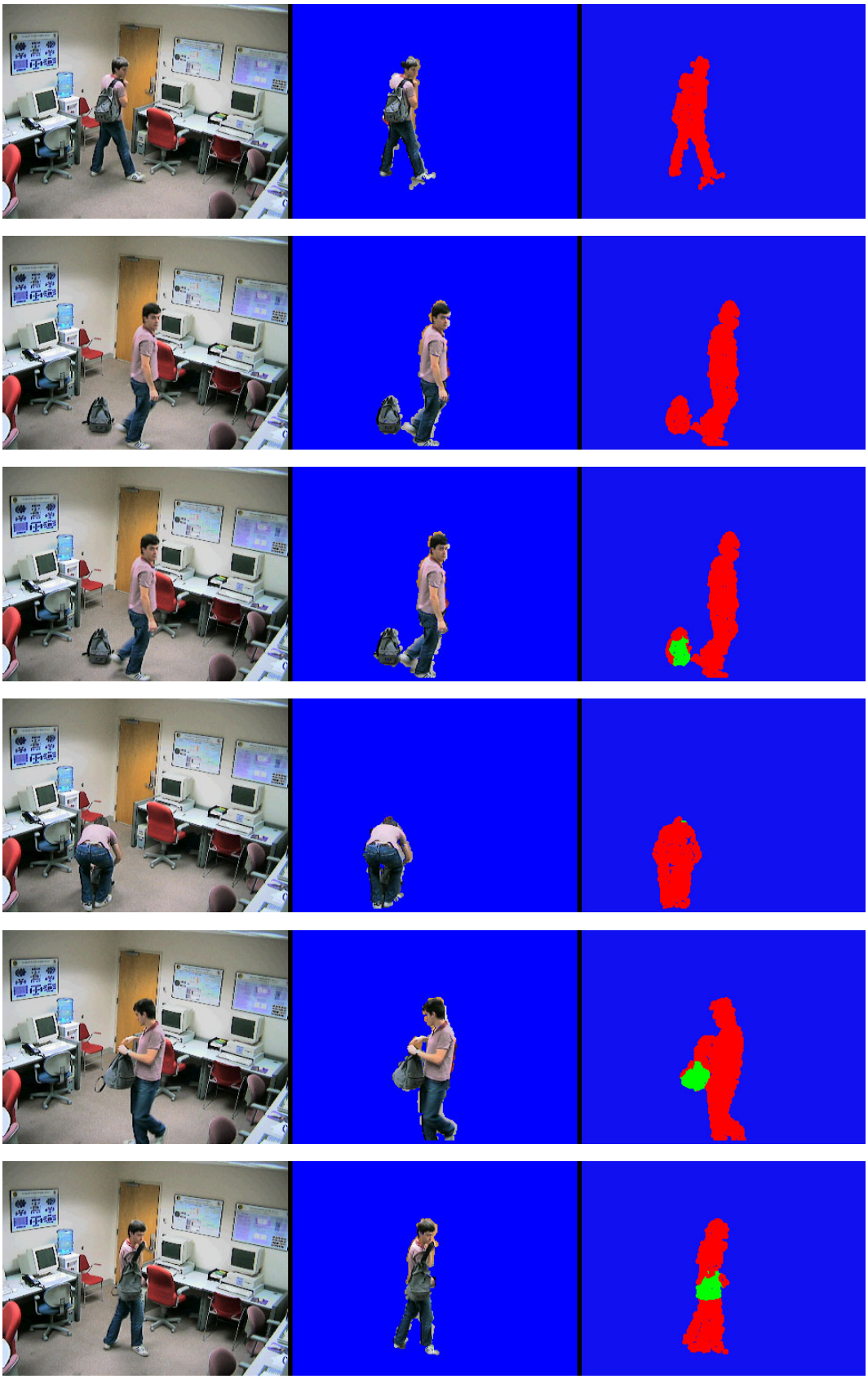


Figure 5.10: Sequence showing a bag being deposited on the floor. Sample frames 264, 664, 664, 563, 472, 416

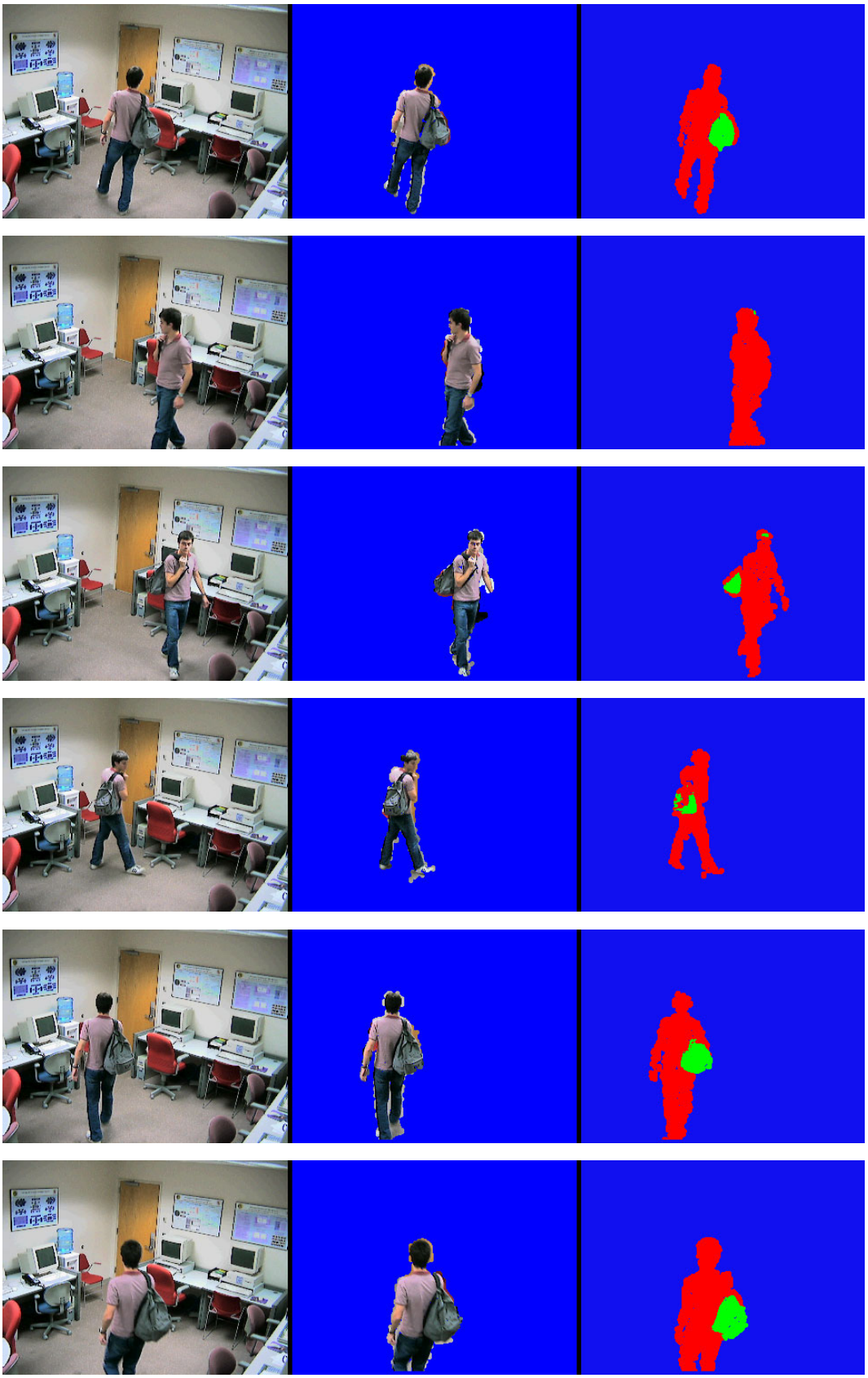


Figure 5.11: Sequence showing a bag being deposited on the floor. Sample frames 384, 343, 311, 264, 220 and 194.

Chapter 6

Summary and conclusions

A system for human and object tracking has been described. An appearance model based on color correlogram is used for tracking and segmenting humans and objects under occlusion. The system performs well for compressed and uncompressed data, and is able to handle common occluding situations that happen in indoor environments. No assumptions are made about human's pose or camera's point of view. In the object tracking case, no assumptions are made about the shape of the object. Since the systems uses an appearance based model, the performance depends on the appearance of objects and humans. If several humans wear similar clothes or the object colors are very similar to the clothes of the person, the performance of the segmentation will degrade. Tracking when individuals are isolated seems to be very robust. Adaptive models together with image normalization allows the system to overcome illumination variations. Under occlusion however, if the models have not been updated for a long period of time, illumination changes may have a more significant effect on segmentation, degrading the results. Object tracking module is more sensitive to illumination variations, since the objects are usually relatively small (therefore have less color information) and the models are initialized using just one frame. In general, segmentation degrades when there are dark colors in shadowed areas or clear colors in highlighted areas.

In the human tracking module, the system assumes that humans are isolated when they enter the scene. If this assumption does not hold the performance will drop, since the models are initialized as humans enter the scene. The system, however, could be adapted to handle this situation. The performance will also degrade if a person merges into a group just after entering the scene, and the few frames where the person is isolated do not contain a significant view of this person (for instance only the head is seen). Segmentation will then be poor and the identities may be lost when the group splits again. The system also assumes that the background subtraction algorithm (after some post-processing stages to merge blobs and fill holes) will provide a single mask for each isolated person. If a blob corresponding to a single person splits into two blobs, the system will be misled and a new model will be initialized for one of the blobs. When a group splits into several subgroups, if one or several humans are completely occluded, the system may make a mistake when deciding which groups are present, since no information is available at that

moment. A way to solve this could be to hold the decision until a significant number of pixels of each person are visible.

The system performance in general is not very sensitive to changes in the parameters of the model (i.e. the distance set where the correlogram is calculated and the number of colors used to quantize the image). For tracking purposes, compared to segmentation, less colors are needed to get good results. In order to optimize the performance one can also change the parameter β that controls how correlogram and histogram information are mixed when doing the segmentation (see (3.16)), although usually $\beta = 0.5$ works well.

For segmentation two methods have been tested, both of them gave good results. In most of the experiments the histogram backprojection method with correlogram correction (explained in section 3.5.1) gave better segmentation results, even though the MAP criterion is supposed to be optimum. This is probably due to the fact that the estimates for the probability density functions are not accurate enough. For the object tracking case, where the models are initialized using just a single frame, probably a kernel density estimation method would have been better. For the human tracking module, since the models are updated at every frame, there are many observations available and the segmentation results would not have been too different between using a kernel density estimation method and histograms. Histograms have the advantage that they are much faster to update. If the objective was to get a very accurate segmentation, more complex method such as the Expectation Maximization (EM), or even Hidden Markov Models (HMM's) could be used. For the human tracking module, one could even maintain several model for each person, initializing a new model when a significant change in appearance is detected.

In general, the system works fairly well when the assumptions made in the designing stage hold. The appearance based model presents a good trade-off between flexibility and robustness: can handle partial occlusion or variations in pose and at the same time is able to do a good segmentation under occlusion. Right now the system does not work in real time, but it could be sped up by several factors using standard optimization techniques.

Some preliminary work was done in order to use the model for tracking humans in the moving camera case. This is an application that should be further studied and we believe that can give good results. More experiments should also be done for testing identification, in order to see how the performance degrades when the number of humans increases. It would also be interesting to test the algorithms in a system with multiple cameras (that could even be in different rooms) to see if the identity is preserved or even if the models can still be used for segmentation. In the object tracking system it would be interesting to see, for example, if the system is able to know in which room the object was picked up and where the same object was deposited. This kind of information is of course very valuable for any surveillance system. Our system could also give useful information to systems willing to perform more accurate human tracking (for example to initialize more complex 3D models), or systems doing activity recognition, where interactions between humans and objects play an important role. Detection of interactions between humans and objects could also be used to summarize a surveillance video or other types of video with a known background. The system could also give significant information to a system managing a teleconferencing environment, where it might be useful to automatically decide which are the more interesting scenes to show to each person.

To conclude, we have presented a system to track humans and objects in an indoor environment. As an appearance model, we believe that correlogram is a powerful representation that could be

used in many other applications. The segmentation method used has also been shown to be very effective. Detection of human-object interactions is a problem that should also be further studied, since it can provide very valuable information for real-time surveillance.

Bibliography

[1] A. Mittal and L. S. Davis. "M2Tracker: A multi-view approach to segmenting and tracking people in a cluttered scene using region-based stereo." In *7th European Conference on Computer Vision (ECCV)*, vol. 1, pp 18-36, Copnehagen, Denmark, May-June 2002.

[2] A. Mittal and D. Huttenlocher. "Scene modeling for wide area surveillance and image synthesis". *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, pp 2160-2167, Hilton Head, South Carolina, USA, June 2000.

[3] J. Orwell, P. Remagnino, and G. A. Jones. "Multi-camera colour tracking". *Proceedings of the 2nd IEEE Workshop on Visual Surveillance*, p. 14, Fort Collins, Colorado, USA, 1999.

[4] J. Krumm, S. Harris, B. Meyers, B. Brumitt, M. Hale, and S. Shafer. "Multi-camera multi-person tracking for EasyLiving". *3rd IEEE International Workshop on Visual Surveillance*, pp 3-10, Dublin, Ireland, July 2000.

[5] S. S.Intille, J. W. Davis, and A. F. Bobick. 'Real-time closed-world tracking". *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp 697-703, 1997.

[6] C. Wren, A. Azarbayejani, T. Darrel, and A. Pentland, "Pfinder: Real-time trackinf of the human body". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 780-785, July 1997.

[7] I. Haritaoglu, D. Harwood and L. S. Davis. "W^4: Real-time surveillance of people and their activities." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp 809-830, August 2000.

[8] I. Haritaoglu, D. Harwood, and L. S. Davis. "W^4S:A real-time system for detecting and tracking people in 2 1/2D". *5th European Conference on Computer Vision*, vol. 22, no. 8, pp. 809-830, Freiburg, Germany, 1998.

[9] S. J. McKenna, S. Jabri, Z. Duric, A. Rosenfeld, and H. Wechsler. "Tracking groups of people". *Computer Vision and Image Understanding*, vol. 80, no. 1, pp. 42-56, October 2000.

[10] A. Senior, A. Hampapur, Y. Tian, L. Brown, S. Pankanti and R. Bolle. "Appearance models for occlusion handling". *Proceedings of the 2nd IEEE International workshop on PETS*, Kauai, Hawaii, USA, December 2001.

[11] L. M. Fuentes, and S. A. Velastin. "People tracking in sureveillance applications". *Proceedings of the 2nd IEEE International workshop on PETS*, Kauai, Hawaii, USA, December 2001.

[12] C. Stauffer, and W. E. L. Grimson. "Adaptive background mixture models for real-time tracking". *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, pp. 246-252, 1999.

[13] T. Horprasert, D. Harwood and L. S. Davis. "A robust background subtraction and shadow detection". *Proceedings of the Asian Conference on Computer Vision*, Taipei, Taiwan, January 2000.

[14] T. Horprasert, K. Kim, and D. Harwood. "Codebook-based adaptive background subtraction for raw compressed videos and a performance evaluation methodology for detection algorithms". 2002 (to be submitted in *European Conference in Computer Vision* 2003).

[15] V Philomin, L. Davis and R. Duraiswami. "Tracking humans from a moving platform". *15th International Conference on Pattern Recognition*, pp. 4171-4179, Barcelona, Spain, September 2000.

[16] C. Bregler, and J. Malik. "Tracking people with twists and exponential maps". *IEEE Conference on Computer Vision and Pattern Recognition*,pp. 8-15, Santa Barbara, California, 1998.

[17] H. Moon, R. Chellappa, and A. Rosenfeld. "3D object tracking using shape-encoded particle propagation". *International Conference on Computer Vision (ICCV)*, pp. 307-314, Vancouver, Canda, July 2001.

[18] H. Sidenbladh, M. J. Black, and D. J. Fleet. "Stochastic tracking of 3D human figures using 2D image motion". *European Conference on Computer Vision*, Dublin, Ireland, pp. 702-718, June 2000.

[19] T. Cham, and J. M. Rehg. "A multiple hypothesis approach to figure tracking" *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 239-245, 1999.

[20] C. Nakajima, M. Pontil, B. Heisele, T. Poggio. "People recognition in image sequences by supervised learning". *Proceedings of IEEE-INNS-ENNS International Joint Conference on Neural Networks*, Como, Italy, July 2000.

[21] Y. Raja, S. J. McKenna, and S. Gong. "Segmentation and tracking using color mixture models". *Asian Conference on Computer Vision*, vol. 1, pp. 601-614, Hong Kong, January 1998.

[22] M. Swain, and D. Ballard. "Color indexing". *International Journal of Computer vision*, vol. 7, no. 1, pp. 11-32, 1991.

[23] B. Funt, and G. Finlayson. "Color constant color indexing". *IEEE Transactions on Pattern Analysis and Machine Intelligence"*, vol. 17, no. 5, pp. 522-529, 1995.

[24] G. D. Finlayson, B. Schiele, and J. L. Crowley. "Comprhensive colour image normalization". *5th European Conference on Computer Vision (ECCV)*, vol. 1, pp. 475-490, Freiburg, Germany, June 1998.

[25] B. Schiele, and A. Pentland. "Probabilistic object recognition and localization". *International Conference on Computer Vision (ICCV)* vol. 1, pp. 177-182, Corful, Greece, September 1999.

[26] B. Schiele, and J. L. Crowley. "Recognition without correspondece using multidimensional receptive histograms". *International Journal on Computer Vision (IJCV)*, vol. 36, no. 1, pp. 31-50, 2000.

[27] A. M. Elgammal, and L. S. Davis. "Probabilistic framework for segmenting people under occlusion". *Proceeding of IEEE 8th Inernational Conference on Computer Vision*, vol. 2, pp. 145-152, Vancouver, Canada, July 2001.

[28] R. M. Haralick, K. Shanmugam, and I. Dinstein. "Textural features for image classification". *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-3, no. 6, pp. 610-621, November 1973.

[29] V. Kovalev, and M. Petrou. "Multidimensional co-occurrence matrices for object recognition and matching". *Graphical Models and Image Processing*, vol. 58, no. 3, pp. 187-197, May 1996.

[30] V. Kovalev, and S. Volmer. "Color co-occurrence descriptors for querying-by-example". *Proceedings of the 5th International Conference on Multimedia Modeling*. pp. 32-38, Lausanne, Switzerland, October 1998.

[31] J. Huang, S. R. Kumar, M. Mitra, and W. Zhu. "Spatial color indexing and applications". *International Journal of Computer Vision*, vol. 35, no. 3, pp. 245-268, 1999.

[32] A. Rao, R. K. Srihari, Z. Zhang. "Geometric histograms: a distribution of geometric configurations of color subsets". *Proceedings of SPIE: Internet Imaging*, vol. 3964, pp. 91-101, January 2000.

[33] K. Fukunaga. "Introduction to Statistical Pattern Recognition". Computer Science and Scientific Computing. Academic Press, New York, 2nd edition, 1990.

[34] M. Turk, and A. Pentland. "Eigenfaces for recognition". *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71-86, 1991.

[35] B. Moghaddam, and A. Pentland. "Maximum likelihood detection of faces and hands". *International Workshop on Automatic Face and Gesture Recognition'* pp. 122-128, 1995.

[36] K. Ohba, and K. Ikeuchi. "Recognition of the multi specularity objects for bin-picking task". *Intelligent Robots and Systens*, pp 1440-1447, 1996.

[37] A. Pentland, R. Picard, and S. Sclaroff. "Photobook: content-based manipulation of image databases". *International Journal of Computer Vision*, vol. 18, no. 3, pp. 233-254, 1996.

[38] M. Stricker and A. Dimai. "Color indexing with weak spatial constraints". *Proceedings of SPIE*, vol. 2670, pp. 29-40, 1996.

[39] J. Smith, and S-F. Chang. "Tools and techniques for color image retrieval". *Proceedings of SPIE*, vol. 2670, pp. 1630-1639, 1996.

[40] R. Rickman, and J. Stonham. "Content-based image retrieval using color tuple histograms". *Proceedings of SPIE*, vol. 2670, pp. 2-7, 1996.

[41] G. Pass, and R. Zabih. "Histogram refinement for content-based image retriebal". *IEEE Workshop on Applications of Computer Vision*, pp 96-102, 1996.

[42] J. Hafner, H. S. Sawhney, W. Equitz, M. Flicknerm, and W. Niblack. "Efficient color histogram indexing for quadratic from distance functions", *IEEE Transactions on Pattern Anaylsis and Machine Intelligence*, vol. 17, no. 7, pp. 729-736, July 1995.

[43] M. Izard, and A. Blake. "Condensation. Conditional density propagation for visual tracking", *International Journal on Computer Vision*, vol. 29, no. 1, pp. 5-28, 1998.

[44] S. Sternberg. "Biomedical image processing", *IEEE Computer*, vol. 16, no. 1, pp. 22-34, January 1983.